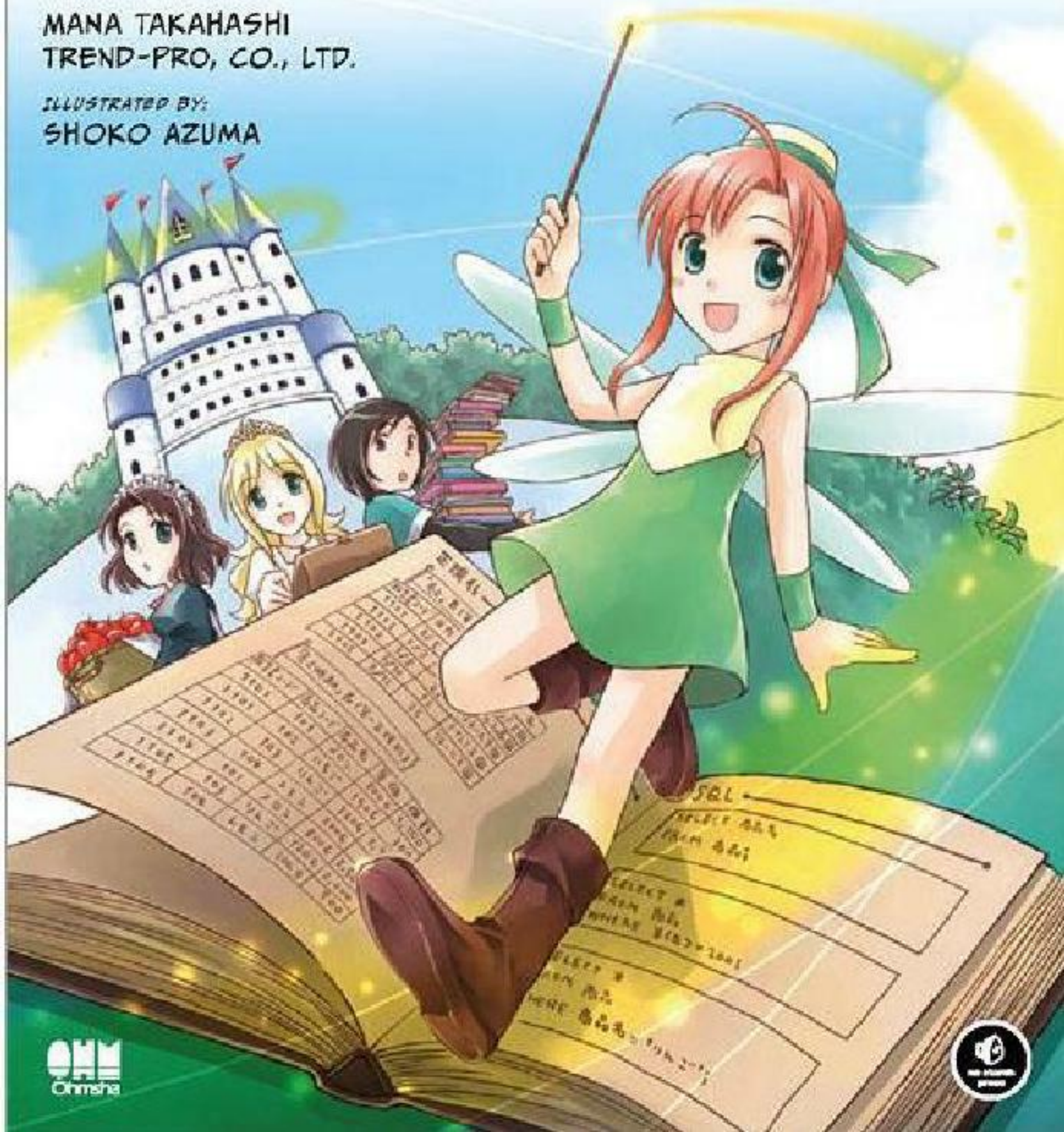


LA GUIA MANGA DE BASE DE DATOS

MANA TAKAHASHI
TREND-PRO, CO., LTD.

ILLUSTRATED BY:
SHOKO AZUMA



CONTENIDO.

Prefacio	I
1. ¿Que es una base de datos?	1
¿por qué necesitamos bases de datos?	2
¿Que pasa en el Reino?	16
Los datos se duplican	16
Los datos pueden entrar en conflicto	17
Datos son difícil de actualizar	18
Una base de datos, ¡es nuestra solución!	19
Cómo utilizar una bases de datos	19
Resumen	21
2. ¿Que es una base de datos relacional?	23
Términos de bases de datos	24
Bases de datos relacionales	34
Tipos de modelos de datos	39
Operaciones de extracción de datos	39
Establecer las operaciones	39
Operaciones relacionales	43
Preguntas	45
La base de datos relacional prevalece!	47
Resumen	48
Respuestas	48
3. ¡vamos a diseñar una base de datos!	49
El modelo E-R	50
Normalizar una tabla	56
¿Qué es el modelo E-R?	74
¿Cómo analizar el modelo E-R?	74
Caso 1: relación uno a uno	74
Caso 2: relación de uno a muchos	75
caso 3: relación muchos a muchos	75
preguntas	76
Normalizar una tabla	78
Preguntas	79
Pasos para diseñar una base de datos	81
Resumen	81
Respuestas	82

4. ¡Vamos a aprende acerca de SQL!	85
Usando SQL	86
Búsqueda de datos utilizando una sentencia SELECT	93
Usando las funciones de agregación	98
Uniando las tablas	101
Creando una tabla	103
Información general de SQL	106
La búsqueda de datos utilizando una sentencia SELECT	106
La creación de condiciones	107
Operadores de comparación	107
Operadores lógicos	107
Patrones	108
Búsquedas	108
Preguntas	109
Funciones de agregación	110
Agregar datos por agrupación	110
Preguntas	111
Buscando Datos sin partícula LA	112
Utilizando una subconsulta	112
Utilizando una subconsulta correlacionada	113
Preguntas	114
Uniando tablas	114
Creando una tabla	115
Insertando, actualizando o eliminando filas	116
Creación de una vista	117
Preguntas	118
Resumen	119
Respuestas	119
5. ¡vamos a utilizar una bases de datos!	125
¿Que es una transacción?	126
¿Que es un bloqueo?	131
Seguridad de base de datos	138
Acelerando cosas con la indexación	143
Recuperacion de desastres	148
Propiedades de las transacciones	153
Atomicidad	153
Consistencia	154
Aislamiento	155
Durabilidad	159
Cuando el desastre azota	161
Tipos de fallos	161
Los puestos de control	161
Preguntas	162

Índices	162
Preguntas	164
Optimización de una consulta	164
Bucle anidado	165
Ordenar fusionar	166
Hash	166
Optimizador	167
Resumen	167
Respuestas	167
6. ¡Bases de datos están en todas partes!	169
Bases de datos en uso	175
Bases de datos y la web	177
Bases de datos distribuidas	183
Procedimientos almacenados y disparadores	185
Bases de datos en la web	194
Usando de procedimientos almacenados	196
Preguntas	196
¿Que es una bases de datos distribuida?	197
Distribución horizontal	197
Distribución vertical	198
Particionamiento de datos	198
Particionamiento horizontal	198
Particionamiento vertical	199p
Previniendo la inconsistencias que impiden cometer en dos fases	199
preguntas	201
replicación de bases de datos	201
sólo lectura	201
replicación habilitada para todos los servidores	202
otras aplicaciones de bases de datos	202
XML	202
bases de datos orientadas a objetos	203
resumen	205
respuestas	205
palabras de clausura	205
apéndice	
de uso frecuente sentencias SQL	207
referencias	209
índice	211

PREFACIO

Las bases de datos son una parte fundamental de casi todos los sistemas de negocios basados en computadoras. Algunos lectores de este libro pueden ser considerados de modo introductorio a las bases de datos en su trabajo rutinario. Otros pueden realmente tener que desarrollar bases de datos reales basadas en sistemas para negocios. Las bases de datos son la tecnología que soporta estos sistemas detrás de las escenas, y su verdadera naturaleza es difícil de entender.

Este libro está diseñado para que los lectores puedan aprender lo básico sobre las bases de datos a través de una historia Manga. al final de cada capítulo, se proporcionan los ejercicios prácticos para la confirmación y ampliación de los conocimientos que ha obtenido. Cada capítulo está diseñado para que los lectores pueden lograr un entendimiento de la tecnología de las bases de datos en el momento que se confirma lo mucho que ellos entienden el contenido.

La estructura de este libro a seguir es:

Capítulo 1 describe por qué usamos las base de datos. ¿Por que es necesaria una base de datos? ¿qué tipo de dificultades va a tener si no utiliza una base de datos? aprenderá los antecedentes de la información de como se están usando los requerimientos de una base de datos.

Capítulo 2 proporciona la terminología básica. Aprenderá acerca de los diferentes modelos de bases de datos y otros términos relacionados con bases de datos.

Capítulo 3 explica cómo diseñar una base de datos, específicamente, una base de datos relacional, el tipo más común.

Capítulo 4 sobre SQL, un lenguaje utilizado para gestionar las bases de datos relacionales. usando SQL que le permite gestionar fácilmente sus datos.

Capítulo 5 explica la estructura del sistema de bases de datos. Desde una base de datos son un sistema a través del cual muchas personas comparten datos, aprenderá cómo puede hacerlo con seguridad.

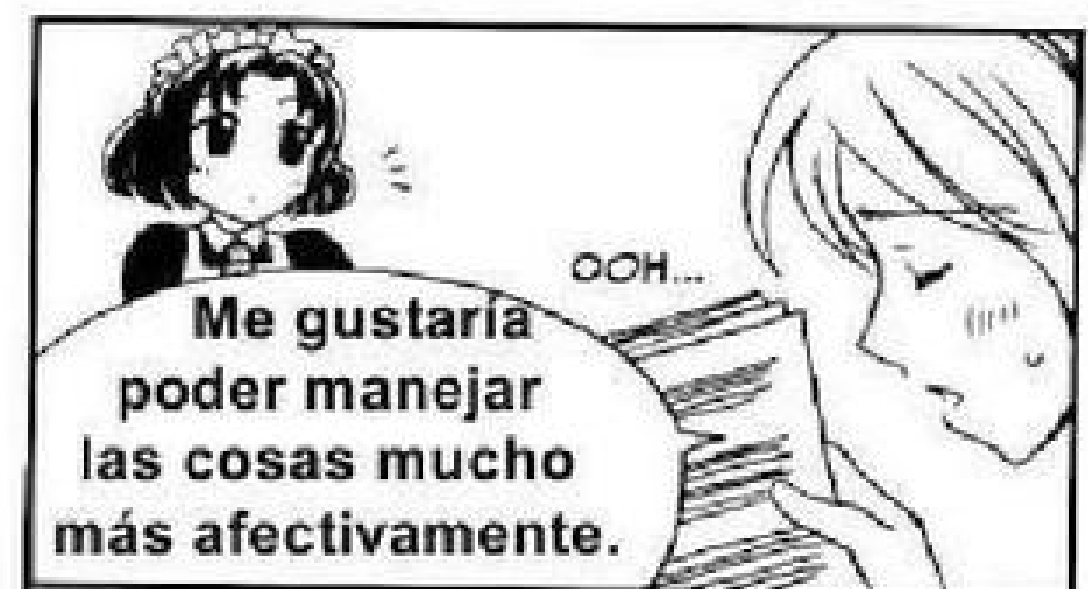
Capítulo 6 proporciona descripciones de las aplicaciones de bases de datos. Aprenderá como son los tipos basados en la web y otros sistemas de bases de datos que son utilizados.

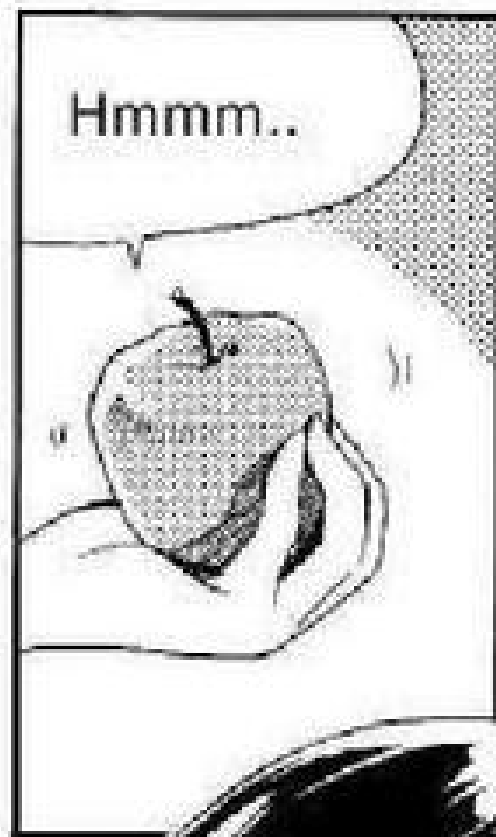
Este libro fue publicado gracias al esfuerzo en conjunto de muchas personas: Shoko Azuma por las caricaturas, TREND-PRO por las producciones y Ohmsha por la planificación, edición y comercialización. Extiendo mi profundo agradecimiento a todos los interesados.

Espero que este libro sea útil a todos los lectores.

MANA TAKAHASHI











"Hemos encontrado
un libro sobre la
tecnología innovadora
en una tierra lejana
que visitamos.

La persona que nos dio
este libro nos dijo que el libro
describe una tecnología secreta
llamada una base de datos.



Nos enteramos de que la
base de datos es un sistema
que permite a todos a compartir,
administrar y utilizar los datos.

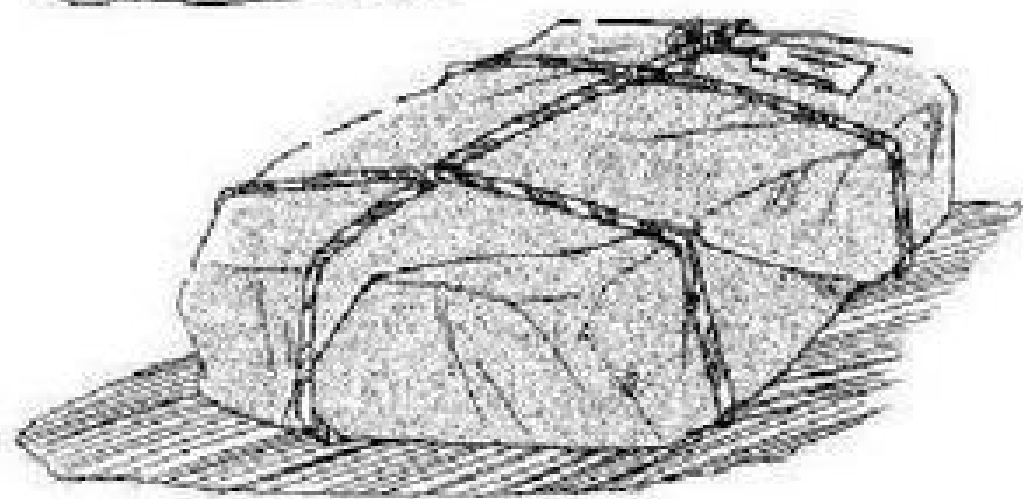
pero, ¿cómo se usa?
depende de quien
lea este libro.



La persona que nos dio este
libro creyó que
el reino de Kod lo debe usar de
manera pacífica.

Ruruna..

abrir este libro, y usarlo
para el bien de
nuestro país ".



¿Qué diablos?

¡Oh, por favor!
tú no sabes nada de lo
de estresada que
estoy sintiendome!

Oh, es
tan viejo ...

Susurro.

Está cerrado.

No puedo abrirlo.

HMM...

Es esta la
llave?

Que había en el sobre ..

si funciona ..

KA-BOOM!

AAAAA!!

?!

HAEE

COUGH
COUGH



¿Bien?

¿dónde
estoy?

¿Y quién
eres tú?

Tú estás en
el castillo de
k... k... Kod

Soy Caín!
- El colaborador
cercano de la
princesa Ruruna
de Kod.

¿Quién eres?

?!

¡¡Vuela!!

ZIP!

¿Un fantasma?

NO!



Bien,
parece inofensiva
hasta ahora ..

¡Si!

Basta de mí!
Ustedes dos se abrió
el libro para aprender
acerca de las
bases de datos ..

¿No es así?

Bueno,
supongo
que sí..

Está bien,
entonces,
vamos
a empezar.

para crear una
base de datos ..

¡Espera
un minuto!

esta es una
pregunta muy
elemental ..

¿Pero que es
una base de datos?

Oh, no sabes
lo que es.

que están manejando
los diversos valores y
números, ¿no es así?

Sí y tengo
muchos
problemas ..

Estoy manejando
valores y los números
relacionados con
los productos,

clientes y ventas
mediante la creación
de archivos en una
base departamental.

ah, va a administrar datos
de una forma coordinada,
por departamento.

HM, HM











¿QUE ESTA PASANDO EN EL REINO?



El reino de Kod actualmente utiliza un sistema de archivos basado en la gestión de sus datos. Pero parece que el sistema actual tiene algunos problemas. ¿Que son ellos, en particular? Hay que mirar el sistema en detalle.

El reino actualmente cuenta con tres departamentos: el departamento mercancías, departamento de negocios en el extranjero, y el departamento de exportación. el departamento de mercancías realiza un seguimiento de toda la fruta producida en el país, el departamento de negocios en el extranjero gestiona los países extranjeros que son los socios de trabajo, de los reinos y el departamento de exportación lleva un registro de la cantidad de fruta de las exportaciones del reino.

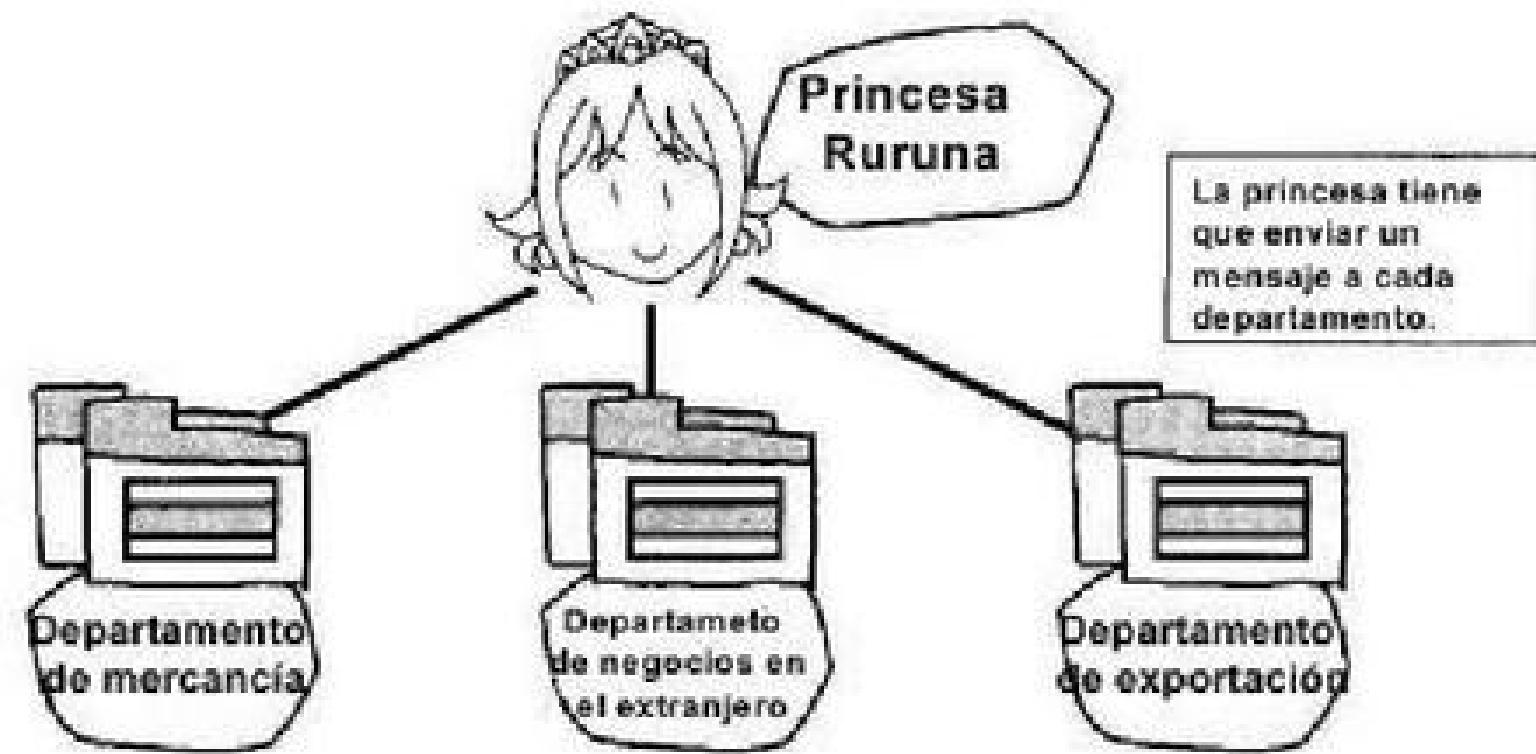


LOS DATOS SE DUPLICAN

princesa Ruruna no está satisfecha con el sistema actual. pero ¿por qué no? cada departamento en el reino gestiona los datos de forma independiente. por ejemplo, el departamento de mercancías y el departamento de exportación de cada uno crear archivos para gestionar los datos de fruta. por lo tanto, los datos se duplican innecesariamente a través de los departamentos. cada departamento debe introducir los datos. almacenar los datos a continuación, imprimir los recibos de confirmación, todo lo cual es un desperdicio. Además, los datos capturados en un departamento en particular, nunca se comparte con eficacia con el otro departamento.



pero eso no es todo, el sistema también crea problemas cuando alguien necesita cambiar los datos. por ejemplo, vamos a suponer que es el precio de las manzanas cambio. Para hacer frente a esta situación. La princesa Ruruna debe notificar a todos los departamentos individualmente que el precio de las manzanas ha cambiado. ¿No es tan incómodo?



LOS DATOS PUEDEN ENTRAR EN CONFLICTO

Puede parecer bastante fácil de notificar a cada departamento que el precio de las manzanas ha cambiado, pero se puede crear un nuevo conjunto de problemas. Que se dice que la princesa Ruruna notifique a los departamentos que el precio de las manzanas ha cambiado. Sin embargo, el departamento de negocios en el extranjero pueden olvidarse de cambiar el precio o el departamento de exportación podría cambiar el precio a 300 g en lugar de 120 g. este tipo de errores resultan en datos contradictorios entre departamentos. Que hace que el contenido de los sistemas de archivos que difieren de las condiciones del mundo real. ¡qué dolor!



DATOS SON DIFÍCIL ACTUALIZAR

El sistema actual no sólo crea conflicto de datos, sino que también hace que sea difícil de responder a los cambios en los negocios. por ejemplo, vamos a decir que el rey quiere poner en marcha un nuevo departamento de turismo. cuando un guía turístico realiza un recorrido por las huertas y analiza las ventas de frutas reinos, el guía tendrá que usar la información más al día de las cifras de ventas. Pero, por desgracia, el sistema actual no permite necesariamente el departamento para acceder a los datos de otros. Ya que los archivos se guardan de forma independiente. Para administrar un negocio de turismo nuevo, la princesa Ruruna tendrá que hacer copias de todos los archivos relevantes para el departamento de turismo!



Archivo para dept. mercancía

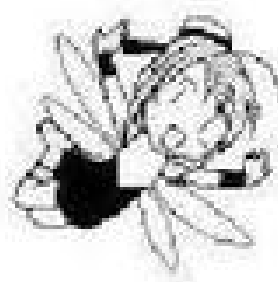
Nombre producto	Precio unidad	
Melón	800G	
Fresa	150G	
Manzana	120G	
Limón	200G	

Archivo para depto. turismo

Nombre producto	Precio unidad	
Melón	800G	
Fresa	150G	
Manzana	120G	
Limón	200G	

Esto a su vez aumenta la cantidad de datos duplicados creados cuando se inicia un nuevo departamento. teniendo en cuenta estas debilidades del sistema actual no es eficiente. Que hace que sea difícil para iniciar nuevos proyectos y responder a los Cambios ambientales.

UNA BASE DE DATOS, ¿ES NUESTRA SOLUCIÓN!



Bien, ¿por qué este sistema es tan ineficiente? ¿Todos los problemas vienen de la gestión de datos separados e independiente lo que deben hacer Ruruna y Caín? Que es correcto -¡ellos deben crear una base de datos! deben unificar los datos de gestión de todo el reino. Yo te mostraré cómo hacer esto en el próximo capítulo.



Administración de datos uniforme asegura que cada departamento tenga la información correcta, ya que cada departamento enviará una consulta a una sola fuente de datos. Esto es un sistema eficiente! evita conflictos de datos, y también elimina datos duplicados que permitan la fácil introducción y la integración de nuevos departamentos.

COMO USAR UNA BASE DE DATOS

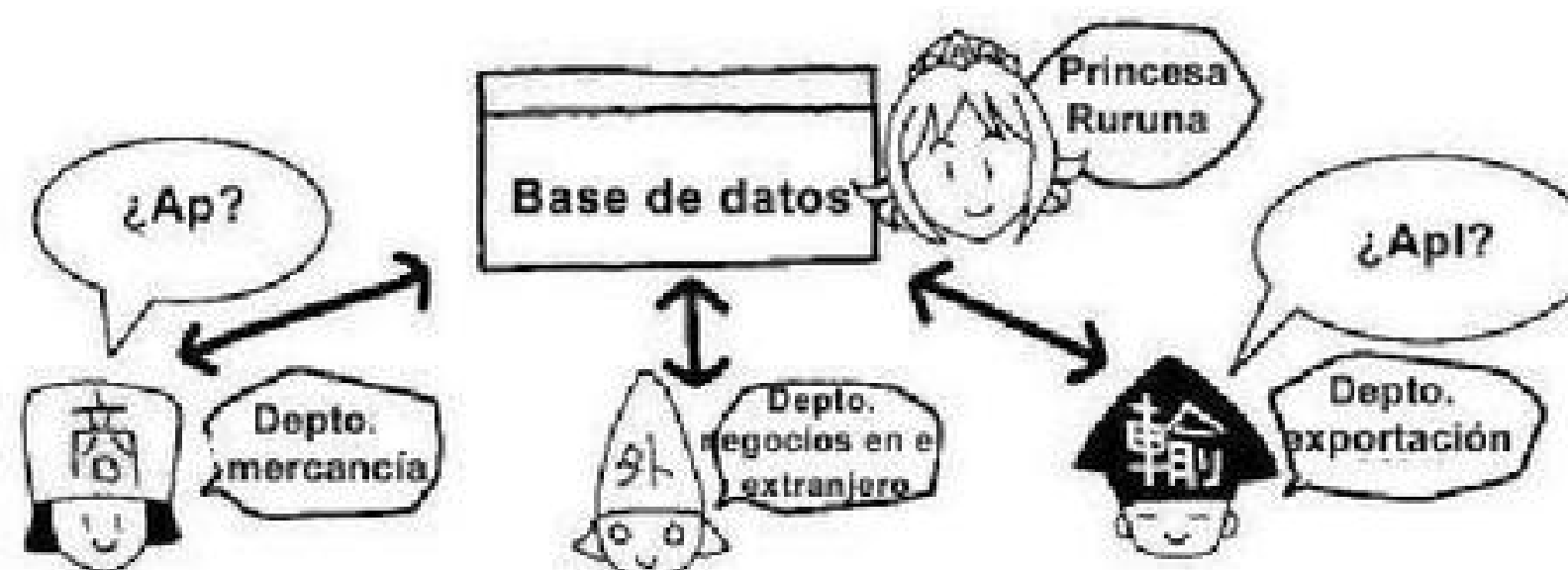
Para introducir y operar una base de datos hay que entender sus retos únicos. Primero la base de datos será utilizada por muchas personas por lo que necesitará una la forma para que ellos de manera fácil puedan entrar y extraer los datos. Necesita ser un método que sea fácil para que todo el mundo pueda usarla.



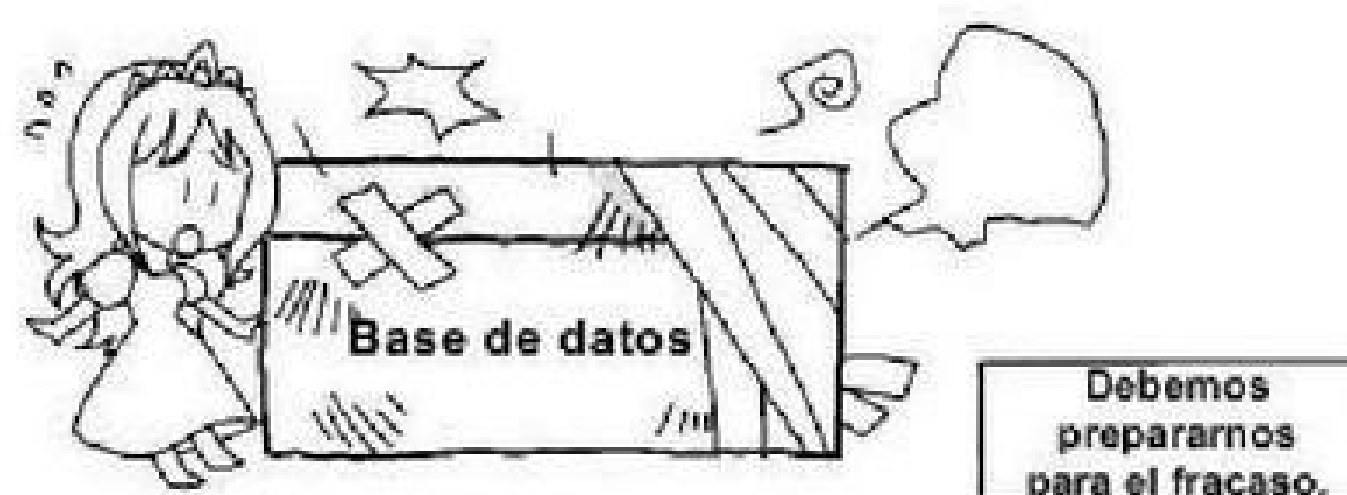
El nuevo sistema también presenta algunos riesgos, por ejemplo, puede que sea posible para el usuario robar o sobrescribir información importante como datos de los salarios, lo cual es confidencial y debe ser protegida por una restricción de acceso. o, por ejemplo, sólo el departamento de exportación deben tener acceso a los datos de ventas de configurar la seguridad de base de datos y el permiso es importante en el diseño de un sistema.



El nuevo sistema puede tener otros problemas, también. La base de datos puede ser utilizada por muchas personas a la vez. Suponiendo que alguien en el departamento de negocios en el extranjero y alguien en el departamento de exportación ambos tratan de cambiar el nombre de una fruta a la vez, el mismo anterior, de una manzana ap, y el segundo, de manzana para apl. si lo hacen, ¿qué va a pasar con el nombre del producto? para una base de datos que serán utilizados por muchas personas, este tipo de problema se debe considerar.



También hay que tener cuidado de no perder ningún dato. Además, el sistema puede ir hacia abajo o un disco duro puede fallar, causando pérdida de datos. La base de datos debe tener mecanismos para recuperarse de estos tipos comunes de fallas.



2

¿QUE ES UNA BASE DE DATOS RELACIONAL?



términos de bases de datos



RAAAMIIINEESSS!!

BANG!!

EEK!





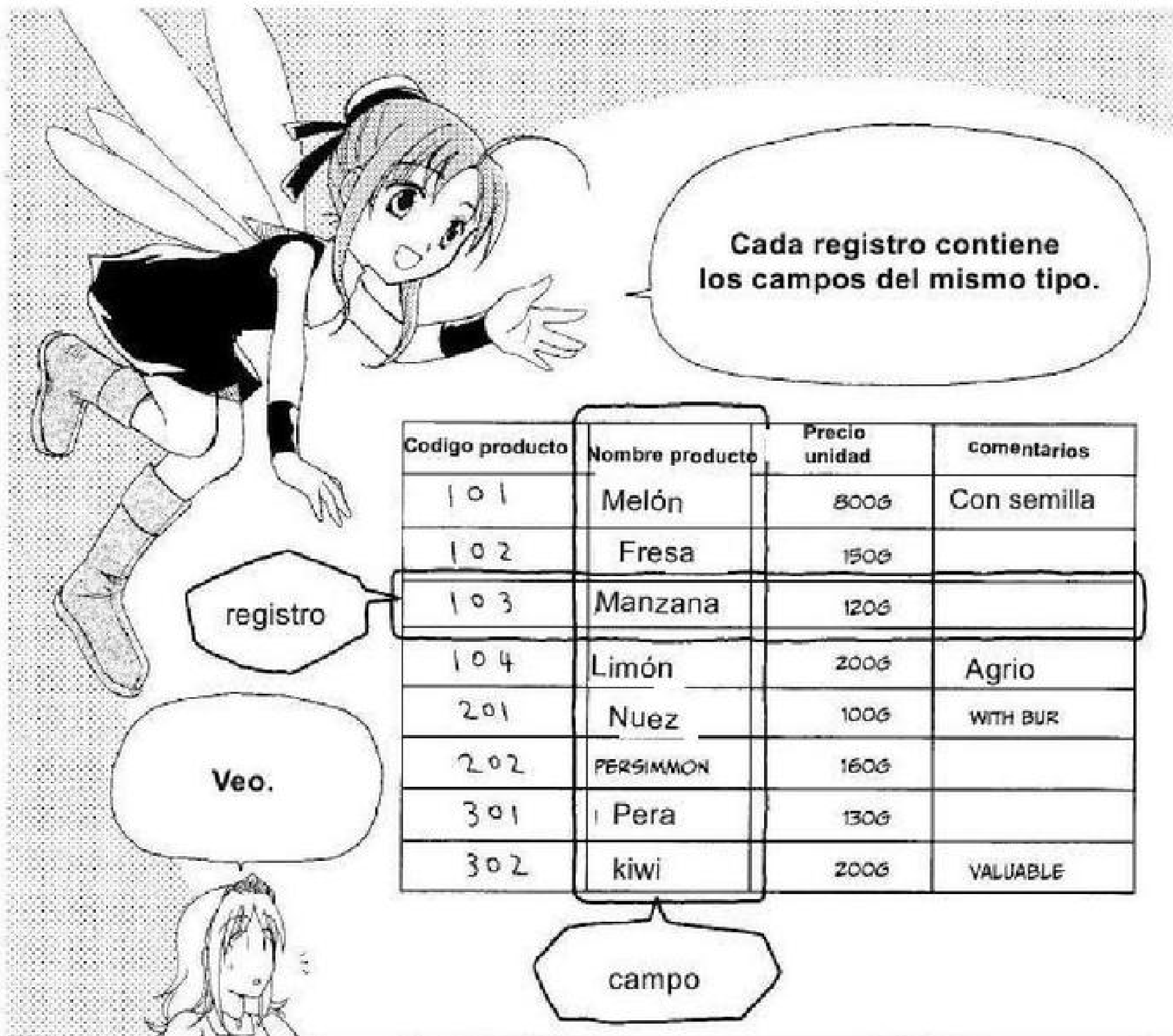


Este es uno de los archivos que está utilizando.

¡Uh-huh!

Codigo producto	Nombre producto	Precio unidad	comentarios
101	Melón	800G	Con semilla
102	Fresa	150G	
103	Manzana	120G	
104	Limón	200G	
201	Nuez	100G	
202	PERSIMMON	180G	









comentario
Con semilla
SOUR
WITH BUR
VALUABLE

Esto no significa que se introduce un espacio ..

es verdaderamente vacío

Si eso es así, no se puede identificar el producto, incluso si nos fijamos en las declaraciones

Nop. .

no puede ser nulo

PRODUCT CODE	PRODUCT NAME	UNIT PRICE	REMARKS
101	MELON	800G	WITH SEEDS
102	STRAWBERRY	150G	
103	APPLE	120G	
104	LEMON	200G	SOUR
201	CHESTNUT	100G	WITH BUR
202	PERSIMMON	160G	
301	PEACH	130G	
302	KIWI	200G	

Nulo

La ausencia de un valor nulo se llama una base de datos en el mundo.

Un nulo es aceptable para los comentarios, pero no para el código de producto, que identifica los datos.

eso es todo por cuanto la base de datos.

¿me entiendes?

Sí..

Bien. Vagamente..

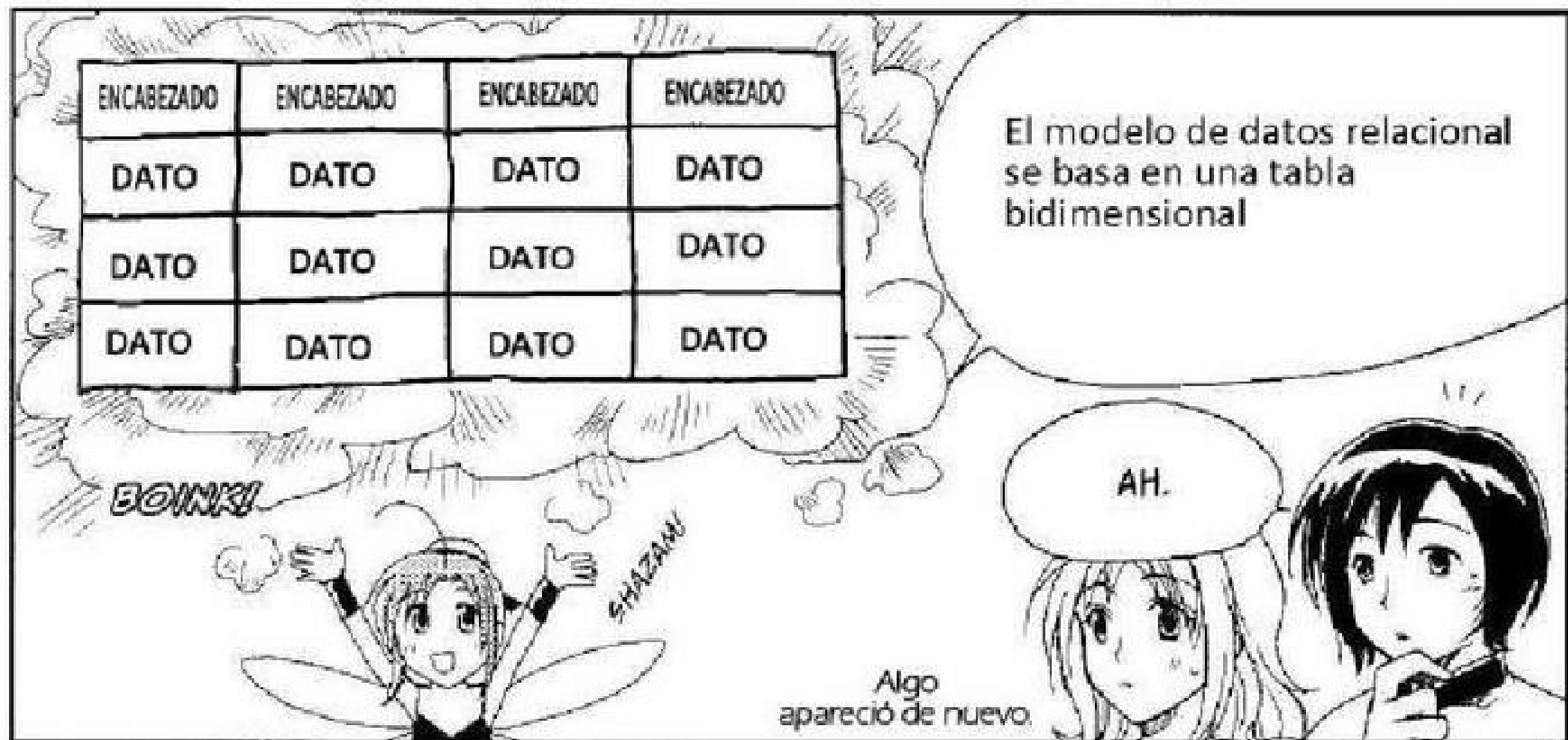
¿nulo? ¿vacío? ¿único?

Murmurar.. Murmurar..





BASE DE DATOS RELACIONAL



Además
un campo se
da a veces un
papel
importante en la
base de datos

Este campo especial
se le llama una llave

key

¿Función
importante?

Si por ejemplo,

El código del
producto en el
archivo que vimos
hace un rato.

El campo desempeña un
papel
importante para
identificar los datos.

Este código
es llamado una
llave primaria.

Yo no sabía
que había
tantos términos.

llave
Primaria

Código de producto
101
102
103
104
201
202
301
302

Bueno, estoy
familiarizada
con las tablas.

Este es un mérito
del
modelo de datos
relacional.

Incluso las personas que
no saben mucho acerca
de las bases de datos
pueden procesar los
datos.

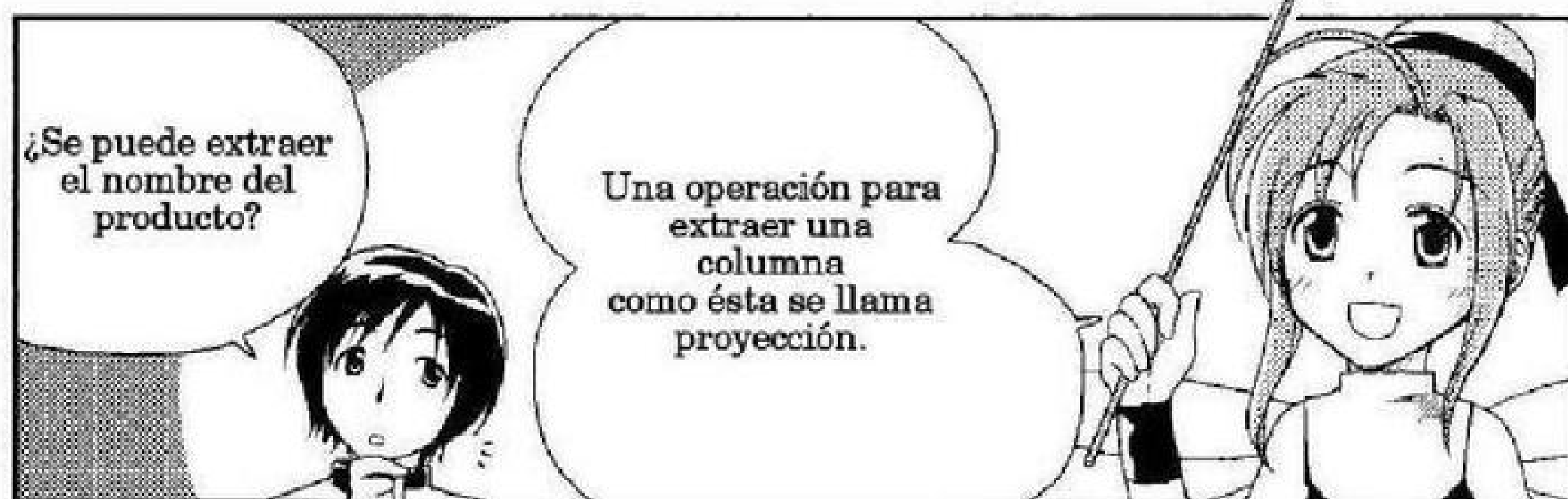
Es fácil de
entender si se
puede procesar
datos mediante
una tabla.

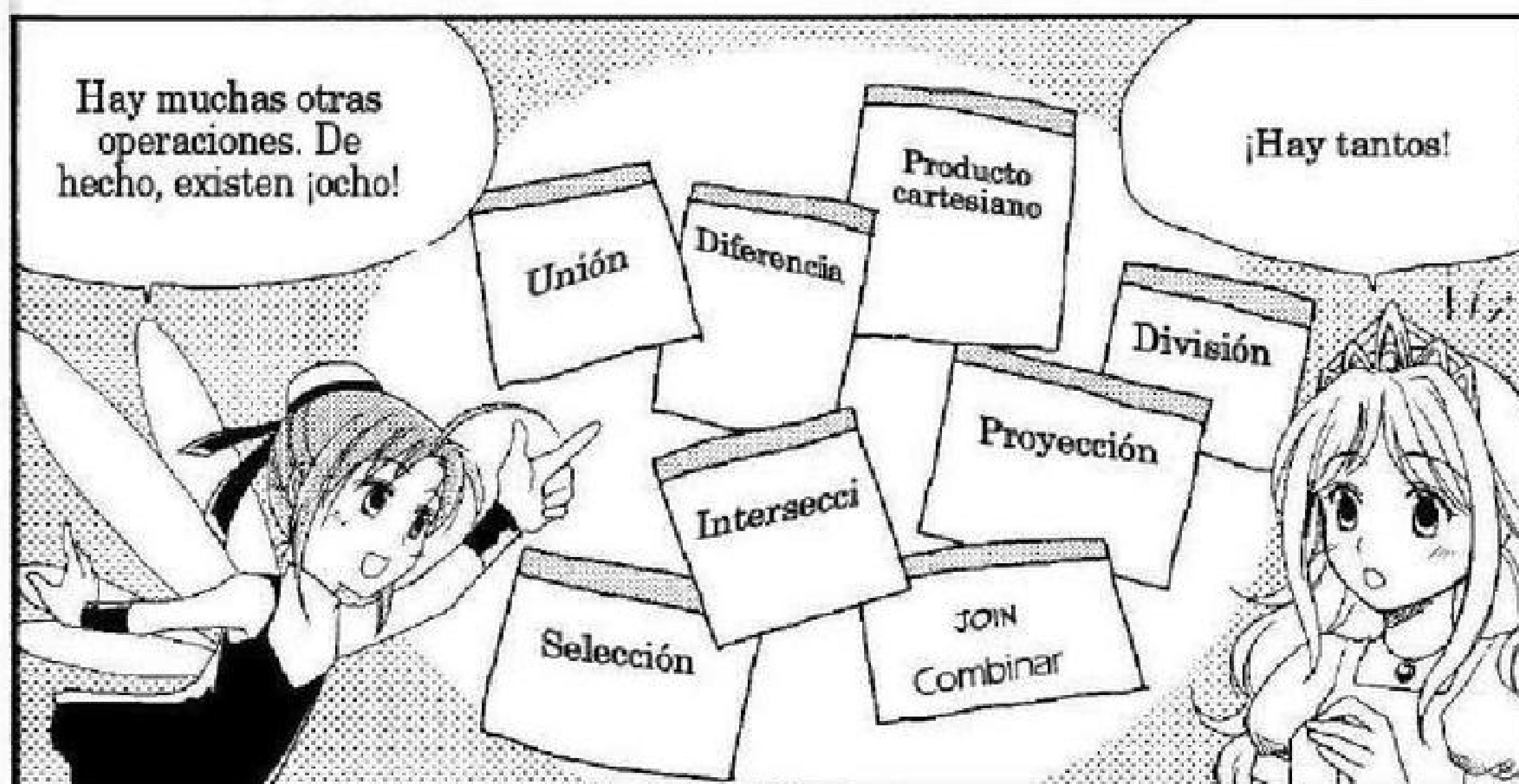


Por ejemplo, echemos un vistazo a la tabla de productos.

Código de producto	Nombre de producto	Precio unitario	Observaciones
101	MELON	8006	Con semillas
102	Fresa	1506	
103	Manzana	1206	
104	Limon	2006	Agrio
201	Caña	1006	Con cubierta espinosa
202	Caqui	1606	
301	melocotón	1306	
302	KIWI	2006	Valioso

Magia!



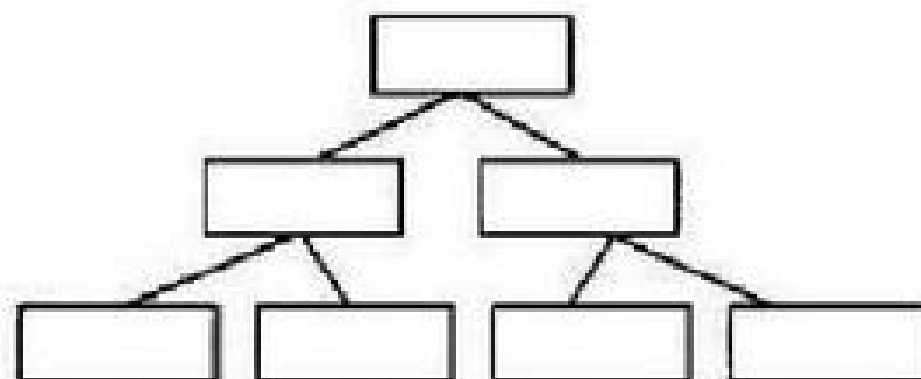


TIPOS DE MODELOS DE DATOS

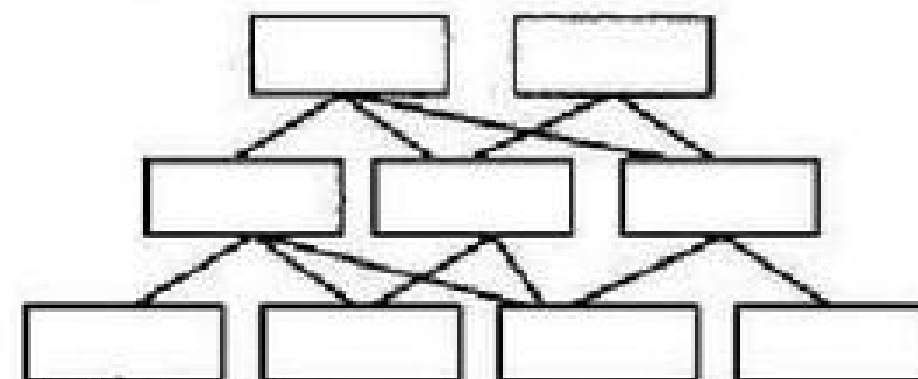


Cuando se utiliza el término base de datos, ¿a qué tipo de base de datos te refieres? Hay muchos tipos disponibles para la gestión de datos. El método de asociación de datos y operación que utiliza una base de datos se denomina modelo de datos. Hay tres modelos de datos de uso común.

Como describí a Ruruna y Caín, el primer tipo es el modelo de datos jerárquico. en el modelo de datos jerárquico, los datos secundarios sólo tiene un pedazo de datos primarios. El segundo tipo es el modelo de datos de red. A diferencia del modelo de datos jerárquico, en el modelo de datos de red, los datos secundarios puede tener múltiples piezas de datos primarios.



Modelo de Datos Jerárquicos



Modelo de Datos de Red

Para utilizar cualquiera de estos modelos, es necesario gestionar los datos, manteniendo la ubicación física y el orden de los datos en la mente. Por lo tanto, es difícil llevar a cabo una búsqueda flexible y de alta velocidad de los datos si se utiliza modelo de datos jerárquico o de red.

El tercer tipo de modelo es el modelo de datos relacional. Una base de datos relacional procesa los datos utilizando el fácil-de-entender el concepto de tabla. Vamos a discutir este modelo en más detalle.

Modelo de Datos Relacional

OPERACIONES DE EXTRACCIÓN DE DATOS

¿Cómo se extrajeron los datos en una base de datos relacional? Puede procesar y extraer datos de una base de datos relacional mediante la realización de operaciones matemáticas estrictamente definidas. Hay ocho principales operaciones que se pueden utilizar, y se dividen en dos categorías --- Operaciones de ajuste y operaciones relacionales.

Establecer operaciones

La unión, diferencia, intersección, producto cartesiano y las operaciones se llaman operaciones establecidas. Estas operaciones funcionan en uno o más conjuntos de fila. En definitiva, determinan que las filas de la entrada aparecen en la salida. Echemos un vistazo a algunos ejemplos utilizando la Tabla de Productos 1 y Tabla de productos 2.

Tabla producto 1

Nombre del producto	Precio Unitario
Melon	800G
Fresa	150G
Manzana	120G
Limon	200G

Tabla producto 2

Nombre del producto	Precio Unitario
Melon	800G
Fresa	150G
Castaña	100G
Caqui	350G

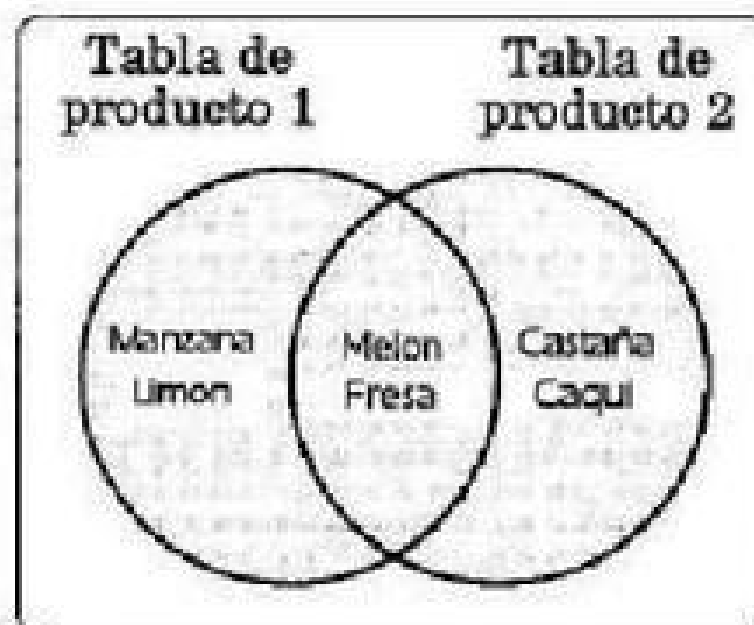
UNION

Llevar a cabo la operación de unión permite extracciones todo producto Tabla 1 y la Tabla del producto 2. El resultado es a continuación.

Nombre del producto	Precio Unitario
Melon	800G
Fresa	150G
Manzana	120G
Limon	200G
Castaña	100G
Caqui	350G

Realizar una operación de unión extrae todas las filas en las dos tablas y las combina.

La siguiente figura muestra lo que los datos de las dos tablas aparecen una vez que una operación de unión se ha realizado. Todas las filas en la tabla de productos 1 y en la tabla de productos 2 se han extraído.

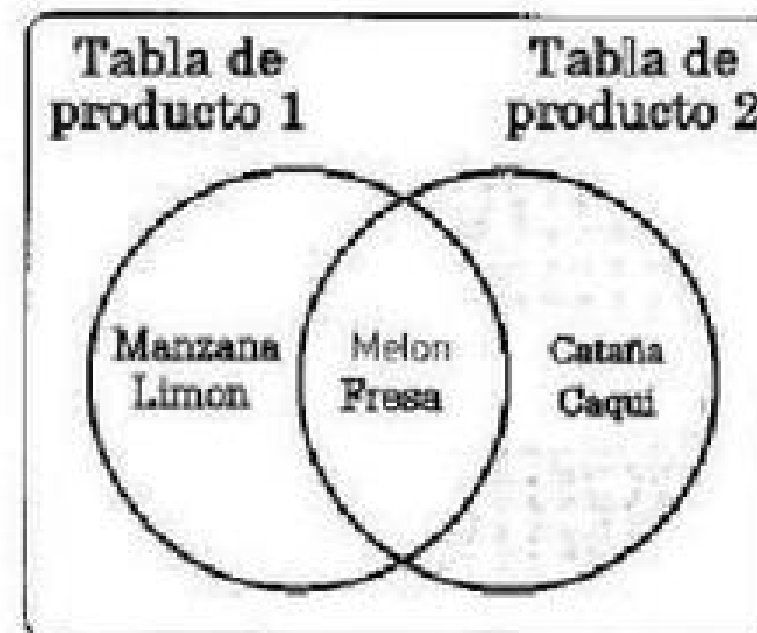
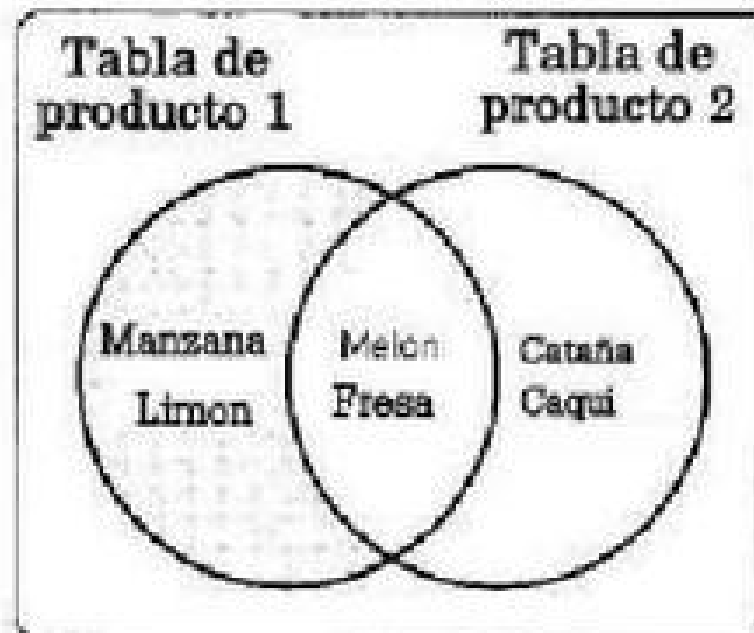


DIFERENCIA

La diferencia es una operación que extrae las filas de sólo una de las tablas. Por ejemplo, una operación de diferencia puede extraer todos los productos de la primera tabla que no están incluidos en el segundo. Los resultados dependen de la tabla que contiene filas para extraer, y la tabla que tiene filas para excluir.

Nombre de Producto	Precio Unitario
Manzana	120G
Limon	200G

Nombre de Producto	Precio Unitario
Caña	100G
Caqui	350G

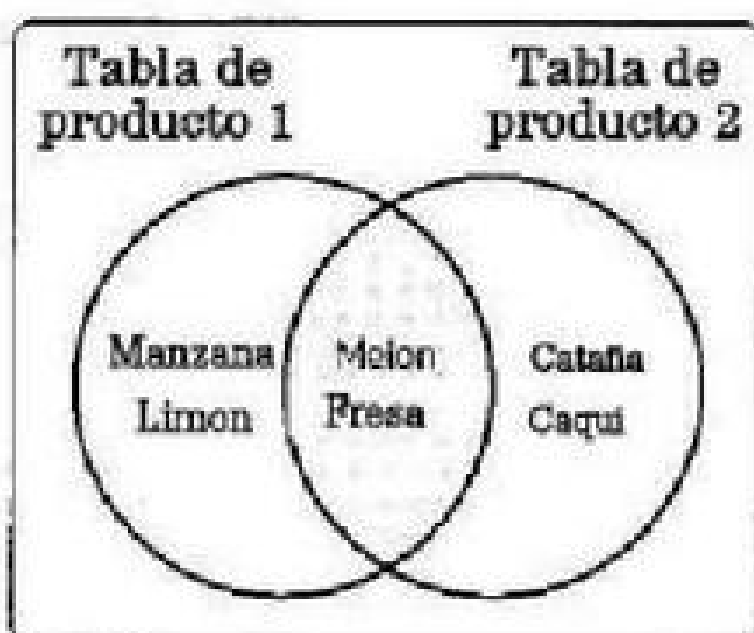


INTERSECCIÓN

También puede extraer los productos que se incluyen tanto en la tabla de producto 1 y en la tabla de productos 2.

Esta operación se denomina una operación de intersección. Este es el resultado de la intersección de la tabla de productos 1 y 2.

Nombre de Producto	Precio Unitario
Melon	800G
Fresa	150G



Producto Cartesiano

El funcionamiento del producto cartesiano es un método que combina todas las filas de las dos tablas. Echamos un vistazo a las tablas de producto y destino de exportación a continuación.

La operación producto cartesiano combina todas las filas de las dos tablas. En este ejemplo, el resultado fue $3 \times 3 = 9$ filas. Observe que los nombres de las columnas (o campos) de las tablas de estas dos no son lo mismo-a diferencia de nuestro ejemplo anterior.

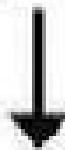
Tabla Producto

Codigo de Producto	Nombre de Producto	Precio Unitario
101	Melon	800G
102	Fresa	150G
103	Manzana	120G

Tabla Destino de exportacion

Codigo Destino de Exportacion	Nombre del Destino de Exportacion
12	El Reino de Minanmi
23	Imperio Alfa
25	El Reino de Ritol

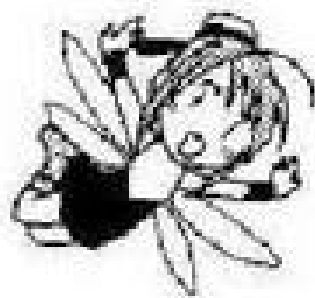
3
Filas



Producto Cartesiano

Codigo de Producto	Nombre de Producto	Precio Unitario	Codigo Destino de Exportacion	Nombre del Destino de Exportacion
101	Melon	800G	12	El Reino de Minanmi
101	Melon	800G	23	Imperio Alfa
101	Melon	800G	25	El Reino de Ritol
102	Fresa	150G	12	El Reino de Minanmi
102	Fresa	150G	23	Imperio Alfa
102	Fresa	150G	25	El Reino de Ritol
103	Manzana	120G	12	El Reino de Minanmi
103	Manzana	120G	23	Imperio Alfa
103	Manzana	120G	25	El Reino de Ritol

3x3=9
Filas



Operaciones relacionales

Una base de datos relacional está diseñada de modo que los datos se pueden extraer por las operaciones de configuración y operaciones relacionales. Echemos un vistazo a las otras cuatro operaciones específicas de una base de datos relacional, llamado relacional operaciones de proyección, selección, conjunta y división.

Proyección

Proyección es una operación que extrae columnas de una tabla. En el ejemplo que se muestra aquí, esta operación se utiliza para extraer sólo los nombres de producto incluidos en la tabla de productos.

Nombre del producto
Melon
Fresa
Manzana
Limon

Piense en proyección como extraer 'verticalmente', como se muestra a continuación.

--	--	--

Selección

La operación de selección extrae dos filas de una tabla.

Nombre del producto	Precio unitario
Melon	800G
Fresa	150G

La selección es como la proyección, sino que extrae filas en lugar de columnas. Selección Extrae datos 'horizontalmente'.

Combinar

La operación de combinación es muy poderosa.
Esta operación se refiere literalmente al trabajo de unión de tablas. Echemos un vistazo a las tablas de abajo como un ejemplo.

Tabla de producto

Código de producto	Nombre del producto	Precio unitario
101	Melon	800G
102	Fresa	150G
103	Manzana	120G
104	Limon	200G

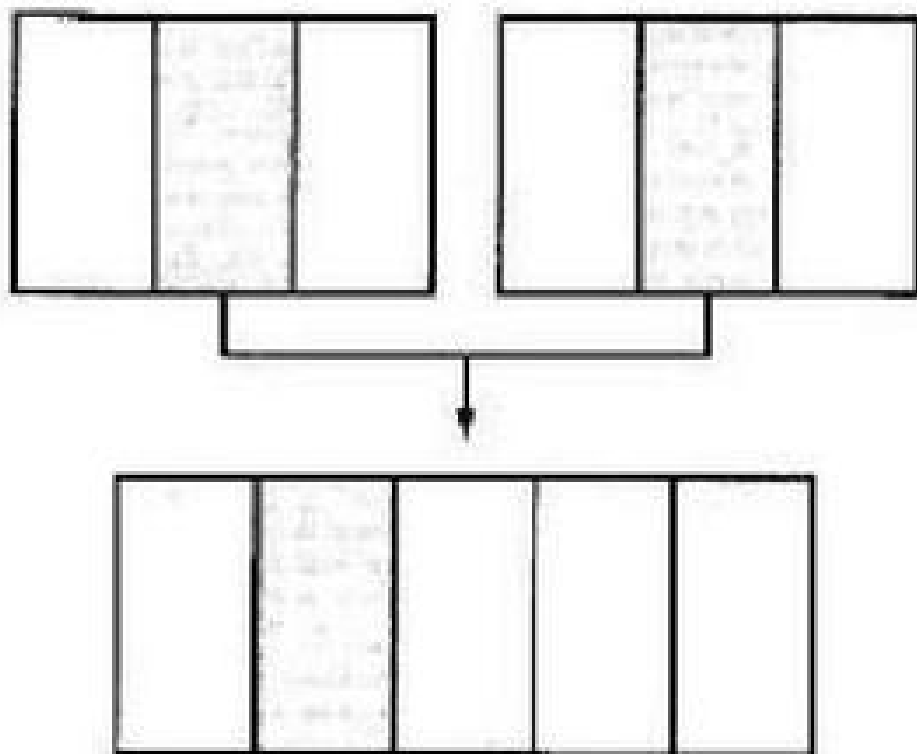
Tabla de ventas

Fecha	Código de producto	Cantidad
11/1	102	1,100
11/1	101	300
11/5	103	1,700
11/8	101	500

Las columnas de códigos de producto en estas dos tablas representan la misma información. El 1 de noviembre, 1.100 fresas (código de producto 102) fueron vendidos. Las tablas de venta no incluyen el nombre del producto, pero sí incluye el código del producto. En otras palabras, la tabla de ventas le permite comprender de qué producto se vendió al hacer referencia al código de producto, cual es la llave primaria de la tabla. El código de producto de la tabla de ventas es una llave foránea. Combinar las dos tablas de modo que la llave foránea se refiere a los resultados de la llave primaria en la tabla siguiente.

Fecha	Código de producto	Nombre del producto	Precio unitario	Cantidad
11/1	102	Fresa	150G	1,100
11/1	101	Melon	800G	300
11/5	103	Manzana	120G	1,700
11/8	101	Melon	800G	500

Esto crea una nueva tabla dinámica de datos de ventas, incluyendo la fecha, código de producto, nombre del producto, el precio unitario y la cantidad. La siguiente figura muestra una unión - el área sombreada representa una columna que aparece en ambas tablas originales.



División

Por último, echemos un vistazo a la división. La división es una operación que extrae las filas de los valores de las columnas coinciden con los de la segunda tabla, pero sólo devuelven las columnas que no existen en la segunda tabla. Veamos un ejemplo.

Tabla de ventas

Código del destino de exportacion	Nombre destino de exportacion	Fecha
12	El Reino de Minanmi	3/5
12	El Reino de Minanmi	3/10
23	Imperio Alfa	3/5
25	El Reino de Ritol	3/21
30	El Reino de Sazanna	3/25

Tabla de destino de exportacion

Código del destino de exportacion	Nombre destino de exportacion
12	El Reino de Minanmi
23	Imperio Alfa

La división de la tabla de ventas por los resultados de la tabla Exportaciones de destino en la tabla siguiente.

Esto le permite encontrar las fechas cuando la fruta se exporta tanto al Imperio Alfa y el Reino de Minanmi.

Fecha
3/5

PREGUNTAS



Ahora, vamos a responder a algunas preguntas para ver qué tan bien usted entiende las bases de datos relacionales. La respuesta está en la página 48.

Q1

¿Cómo se llama la llave en referencia a una columna en una tabla diferente en una base de datos relacional?

Q2

La tabla siguiente muestra información sobre los libros. Cual de ellos puede utilizar como clave principal? El ISBN es el Número Internacional Normalizado para Libros. Un número de identificación único dado a cada libro publicado. Algunos libros pueden tener el mismo título.

ISBN	Nombre del Libro	Nombre del Autor	Fecha de publicacion	Precio
------	------------------	------------------	----------------------	--------

Q3

¿Cómo se llama la operación utilizada para extraer datos?

Código del destino de exportación	Nombre destino de exportación	→	Código del destino de exportación	Nombre destino de exportación
12	El Reino de Minanmi		25	El Reino de Ritol
23	Imperio Alfa			
25	El Reino de Ritol			
30	El Reino de Sazanna			

Q4

¿Cómo se llama la operación utilizada para extraer datos?

Código del destino de exportación	Nombre destino de exportación		Código del destino de exportación	Nombre destino de exportación
12	El Reino de Minanmi		15	El Reino de Paronu
23	Imperio Alfa		22	El Reino de Tokanta
25	El Reino de Ritol		31	El Reino de Taharu
30	El Reino de Sazanna		33	El Reino de Mariyon
		↓		
Código del destino de exportación	Nombre destino de exportación			
12	El Reino de Minanmi			
15	El Reino de Paronu			
22	El Reino de Tokanta			
23	Imperio Alfa			
25	El Reino de Ritol			
30	El Reino de Sazanna			
31	El Reino de Taharu			
33	El Reino de Mariyon			

Q5

¿Cómo se llama la operación utilizada para extraer datos?

Código de destino de exportación	Nombre de destino de exportación
12	El Reino de Minanmi
23	Imperio Alfa
25	El Reino de Ritol
30	El Reino de Sazanna

Código de destino de exportación	Fecha
12	3/1
23	3/1
12	3/3
30	3/5
12	3/6
25	3/10



Código de destino de exportación	Fecha	Nombre de destino de exportación
12	3/1	El Reino de Minanmi
23	3/1	Imperio Alfa
12	3/3	El Reino de Minanmi
30	3/5	El Reino de Sazanna
12	3/6	El Reino de Minanmi
25	3/10	El Reino de Ritol

¡LA BASE DE DATOS RELACIONAL PREVALECE!

En una base de datos relacional, se puede usar ocho operaciones diferentes para extraer los datos. Los resultados extraídos se tabulan. Si se combinan las operaciones que se explican en esta sección, puede extraer los datos para cualquier propósito. Por ejemplo, puede utilizar el nombre y el precio de un producto para crear datos agregados brutos de venta de la misma. Base de datos relacional son populares porque son fáciles de entender y proporcionar un procesamiento de datos flexible.

3

¡VAMOS A DISEÑAR UNA BASE DE DATOS!



El modelo E-R

RUSTLE

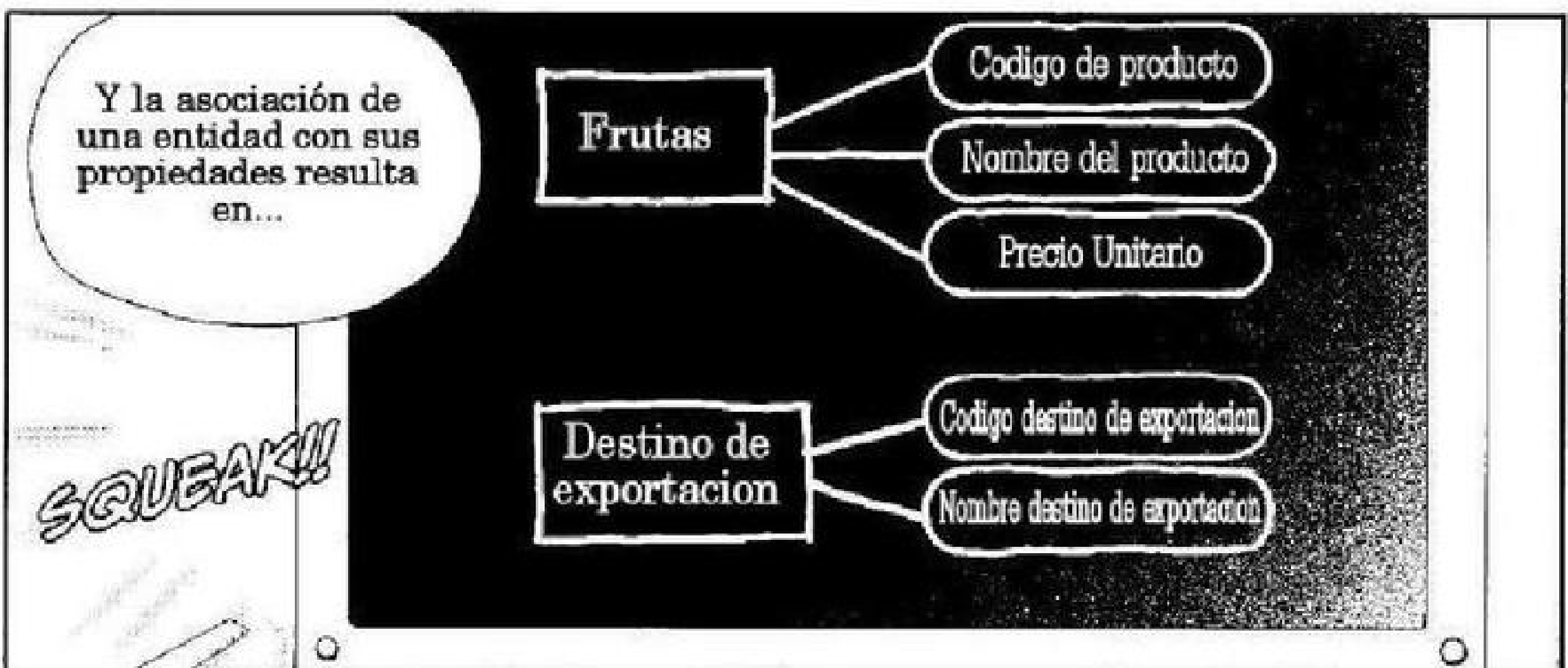
RUSTLE

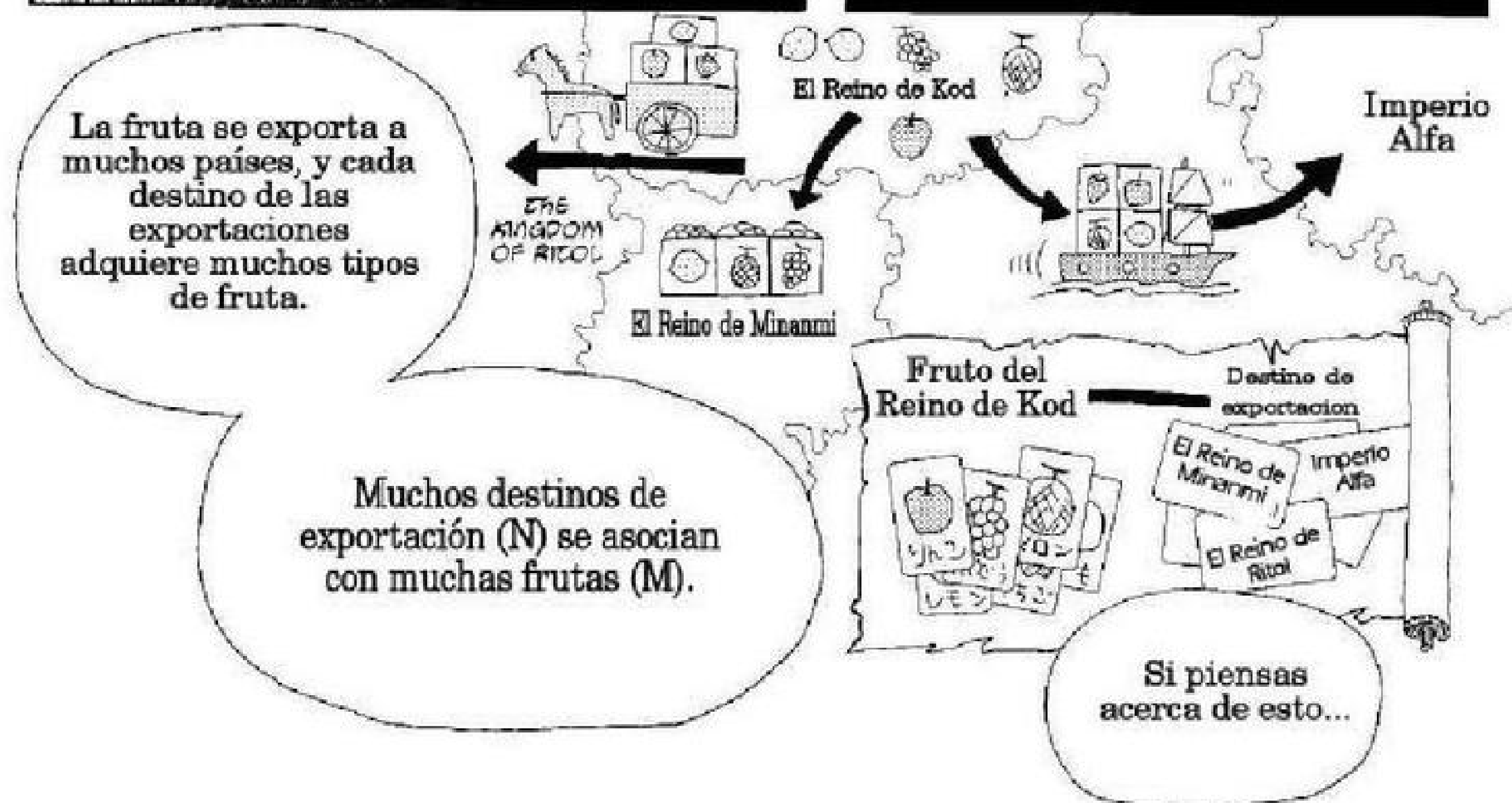
¿Cain?
¿Dónde estás?

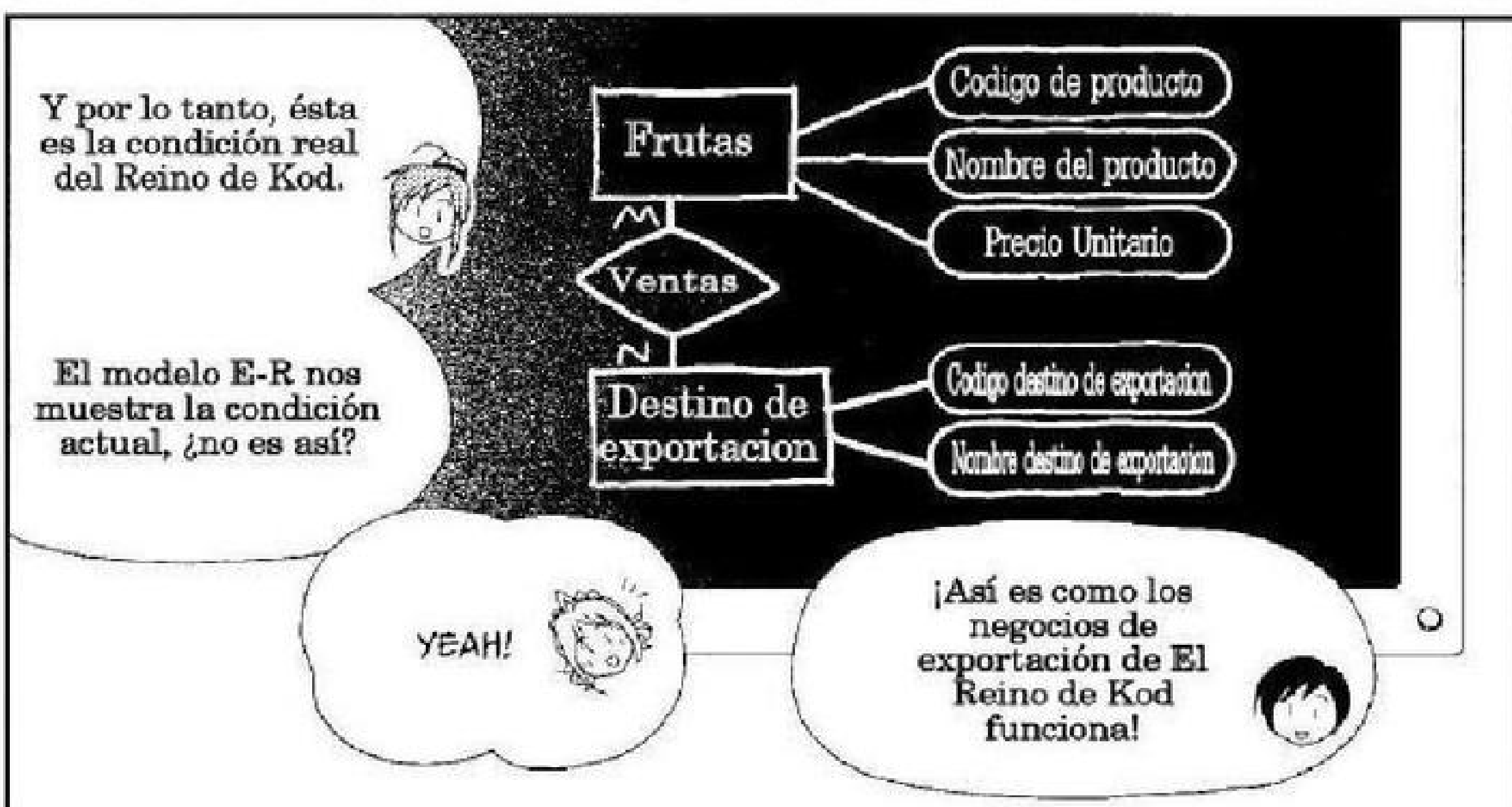
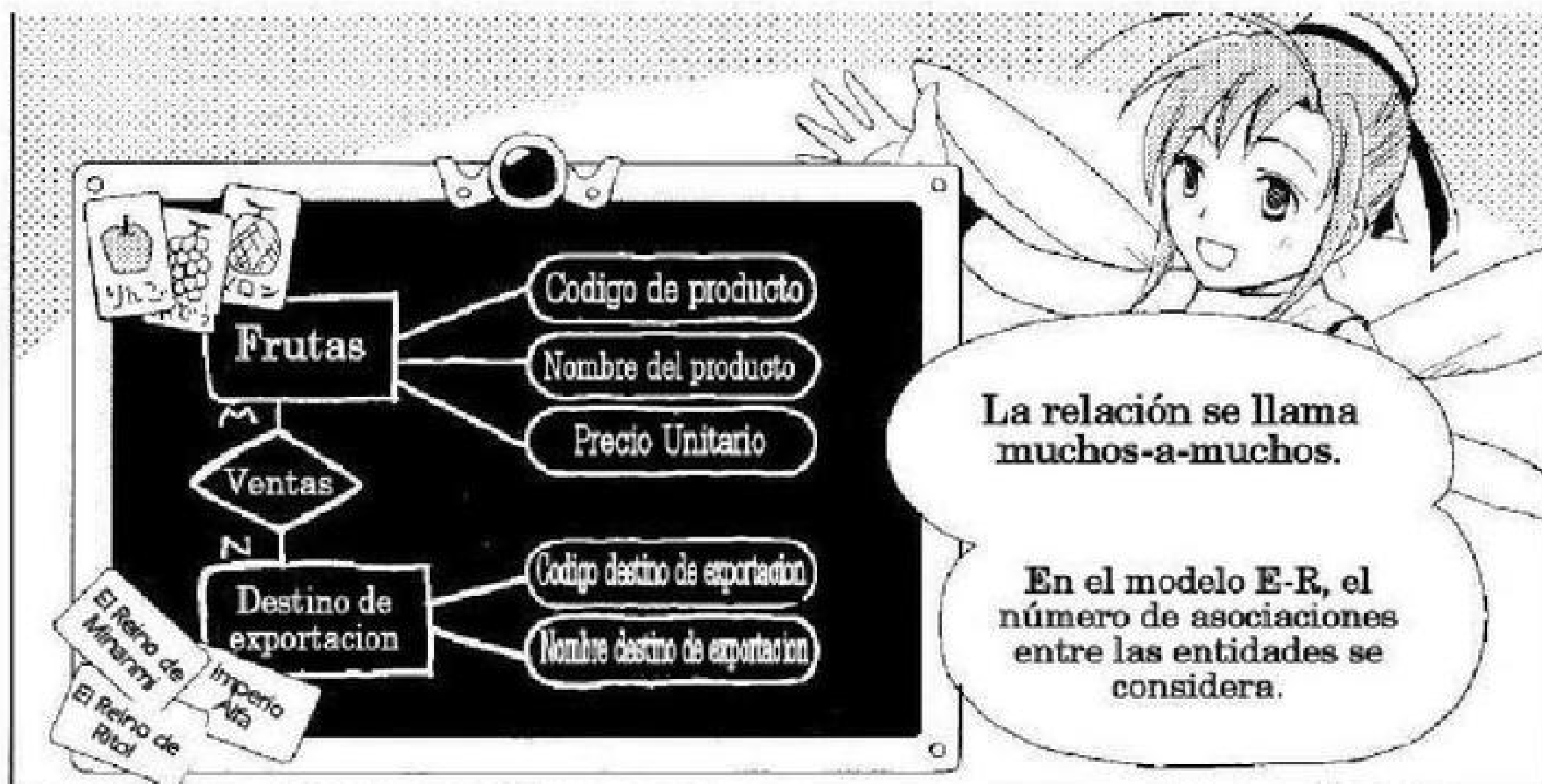












Normalizando una Tabla

Es difícil empezar a crear una base de datos.

¡Eso es! Lo primero que debemos hacer es analizar el estado actual. Eso es muy

Ahora que sabes acerca de la condición actual del Reino de Kod...

Vamos a considerar el diseño de una base de datos real.

¡Si-i-i-i!

Sorprendida

MR. CAN?

SHHH!

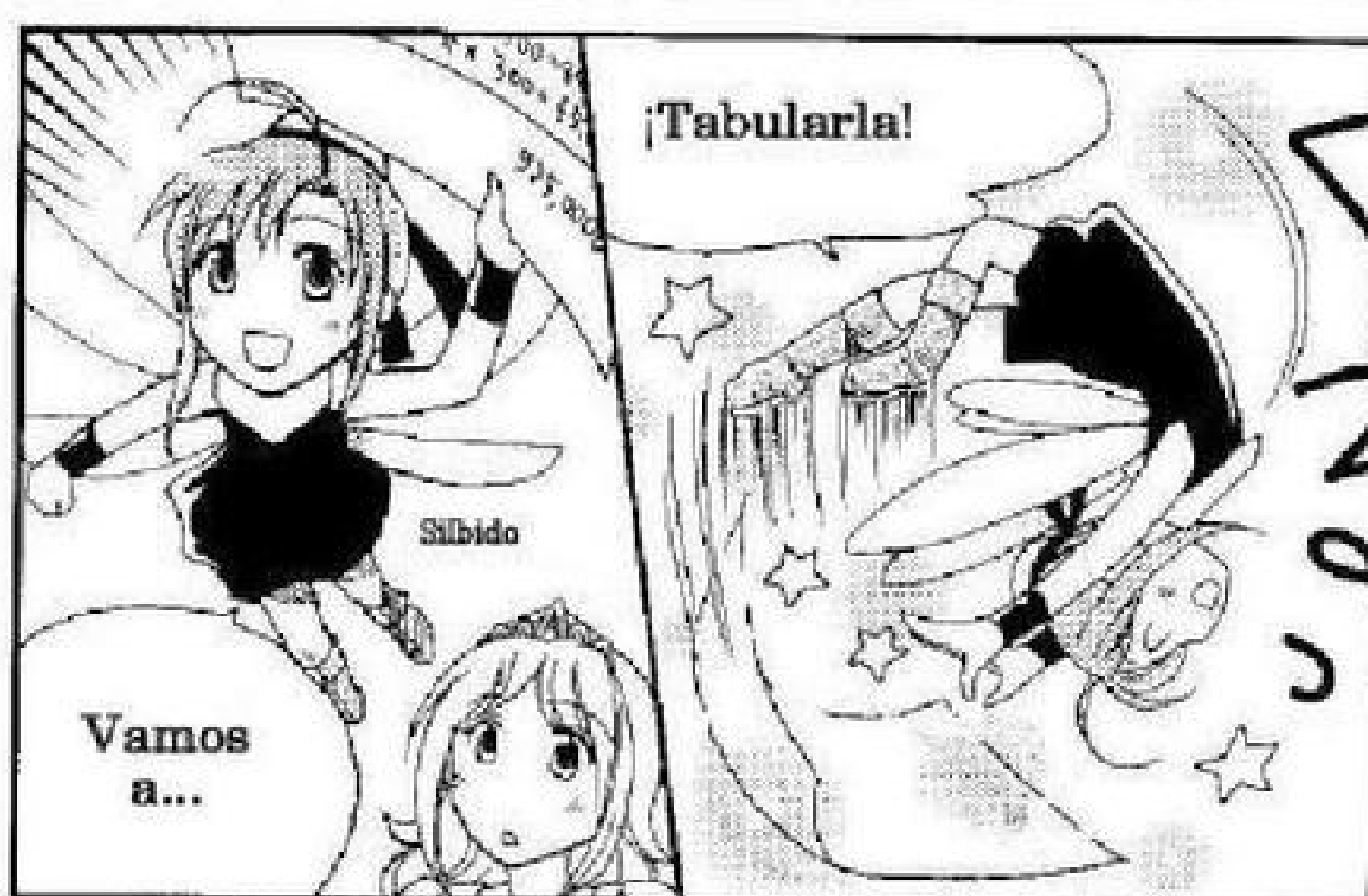
¿Princesa?

HMM...

Lo encontré.

¡Bucear en!

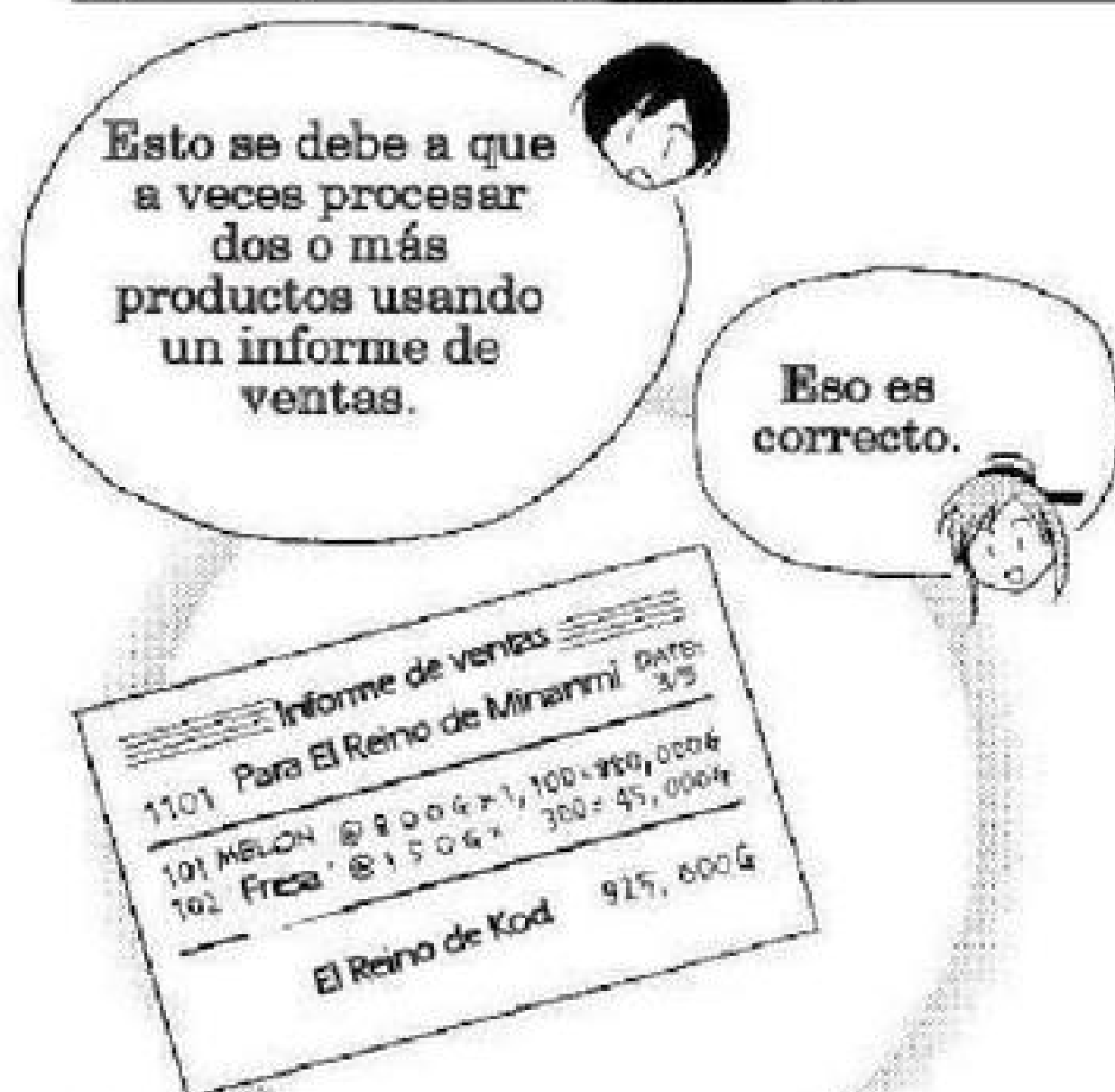
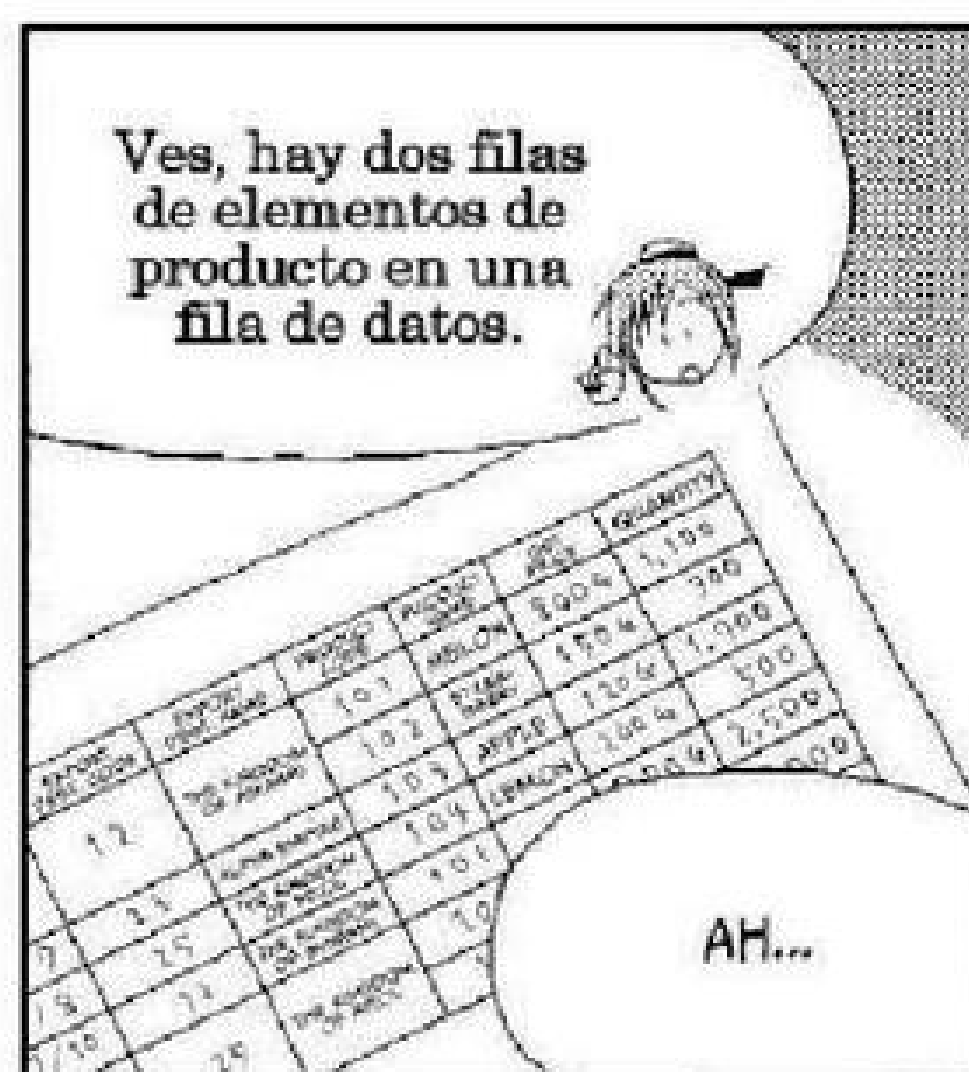
Informe de Ventas
1101 Para El Reino De Minami Fecha: 3/5
101 MELON @800G x 1,100 = 880,000G
102 Fresa @150G x 300 = 45,000G
El Reino De Kod TOTAL 925,000G

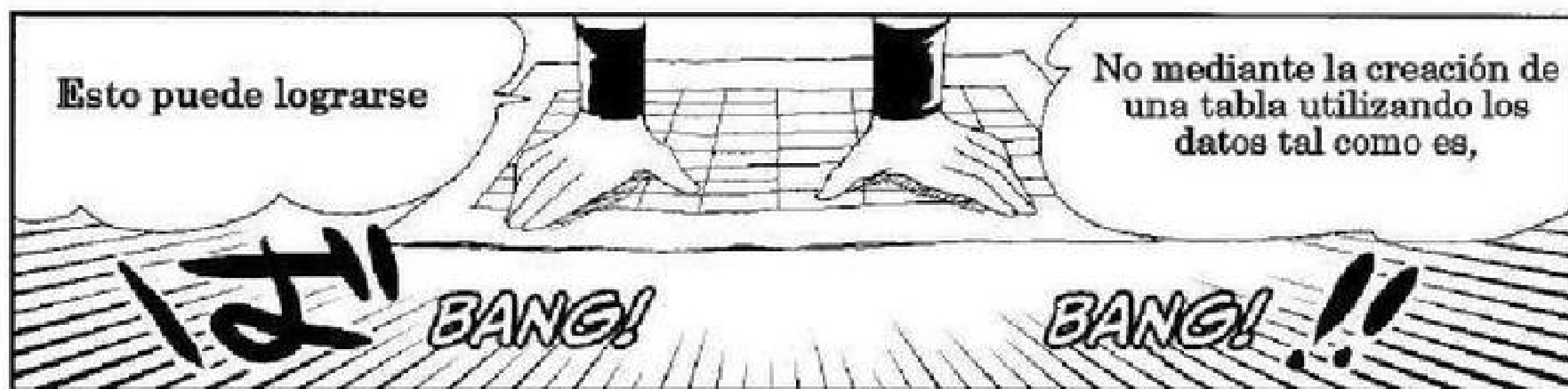


Aquí tienes.

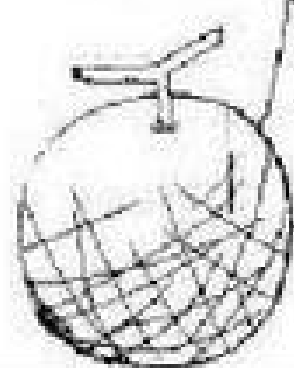
Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación	Código de producto	Nombre de producto	Precio Unitario	Cantidad
1101	3/5	12	El Reino de MINAMI	101	MELON	800G	1,100
				102	Fresas	150G	300
1102	3/7	23	Imperio Alfa	103	Manzana	120G	1,700
1103	3/8	25	El Reino de RITOL	104	Limon	200G	500
1104	3/10	12	El Reino de MINAMI	101	MELON	800G	2,500
1105	3/12	25	El Reino de RITOL	103	Manzana	120G	2,000
				104	Limon	200G	700

Tabla creada a partir de informe de ventas





Supongamos que desea elevar el precio por unidad del melón por 20G.



Si utiliza esta tabla como es,

Usted tiene que encontrar todas las filas de melón y corregir el precio unitario.



Pero, si tienes una tabla que relaciona exclusivamente a los productos,

Puedes corregir el precio en un solo lugar: una fila en la tabla de productos.

Tabla de productos

MELON	800G
Fresa	150G
Manzana	120G
Limon	200G

¡Sólo aquí!

¡Fácil!



Ningún conflicto se produce, ¡incluso si se olvida de corregir las otras filas!
¿No es genial?

Ya sabes, utilizando una sola tabla hace que sea fácil de olvidar corregir los datos.

Ya veo. Visto desde esta perspectiva, no es conveniente.

Se divide la tabla para evitar que conflictos de datos como estos ocurran...

Se llama normalización.

Normalización, normalización

¿Murmurando, otra vez?

¡Sí, esto es importante!

Entonces, ¿qué se supone que debo hacer?

En primer lugar...



Por lo tanto, lo voy a dividir en...

Una tabla con la fecha, el código de destino de las exportaciones, y el nombre del destino de las exportaciones...

Y otra tabla con el código del producto, nombre del producto, el precio unitario y la cantidad.

Tabla ventas (primera forma normal (1))

Código de Informe	Fecha	Código destino de exportación	Nombre destino de exportación
1101	3/5	12	El Reino de MINAMI
1102	3/7	23	Imperio Alfa
1103	3/8	25	El Reino de RITOL
1104	3/10	12	El Reino de MINAMI
1105	3/12	25	El Reino de RITOL

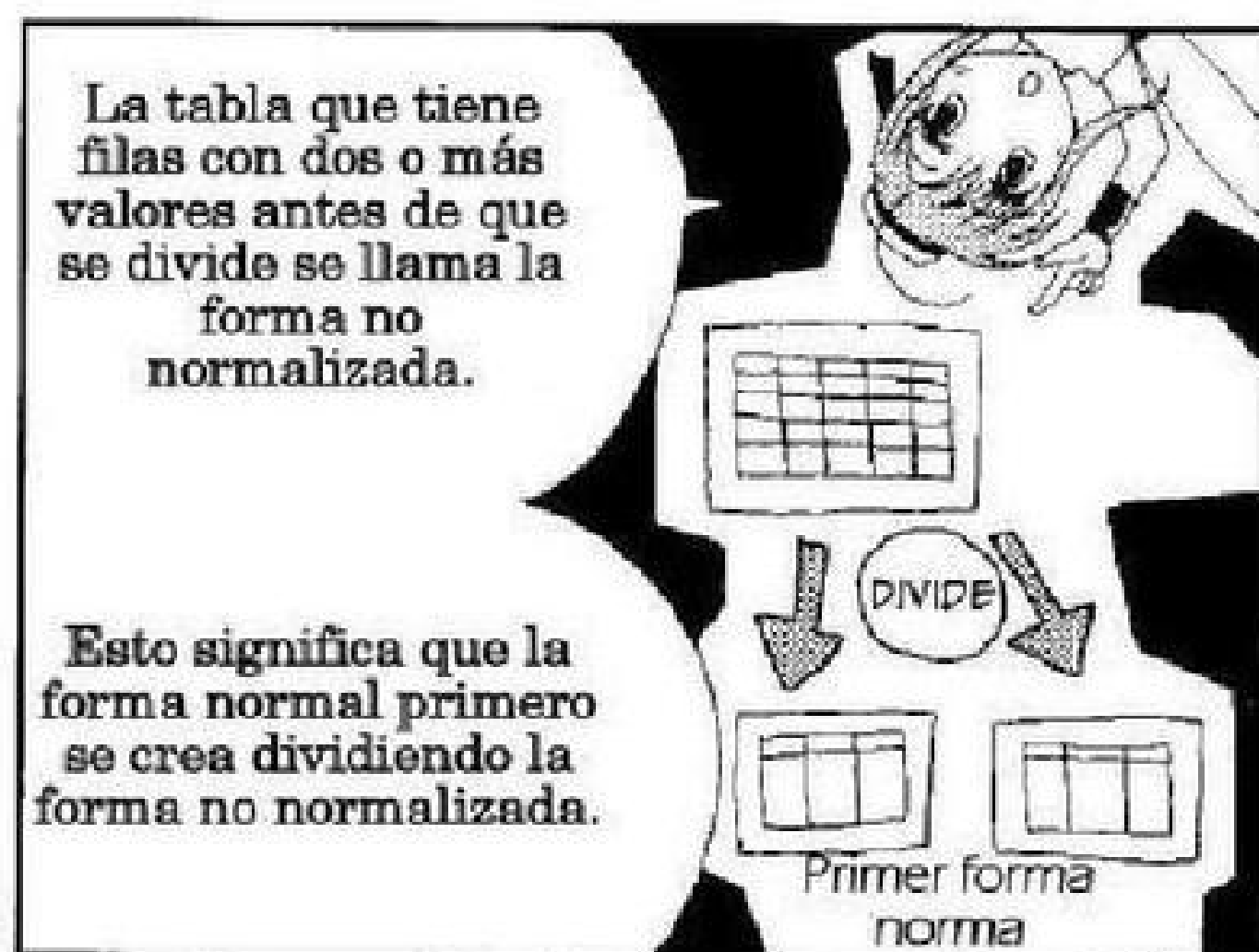
Sin embargo, el código de informe está disponible en ambas tablas, ¿no es así?

Tabla ventas (primera forma normal (2))

Código de Informe	Código de producto	Nombre de producto	Precio unitario	Cantidad
1101	101	MELON	800G	1,100
1101	102	Fresa	150G	300
1102	103	Manzana	120G	1,700
1103	104	Limon	200G	500
1104	101	MELON	800G	2,500
1105	103	Manzana	120G	2,000
1105	104	Limon	200G	700

HUH.

Sí, de esa manera se puede identificar si existe una relación entre las dos tablas.





Código de informe	Código de producto	Nombre de producto	Precio Unitario	Cantidad
1101	101	MELON	800G	1,100
1101	102	Fresa	150G	300

Tabla declaración de ventas (primera forma normal (1))

Todavía no se puede administrar productos con esta tabla.

AIEEE!! ¿Por qué?





Estas son las tablas que resultan de la división de la primera forma normal (2) en dos.

Tabla de productos (segunda forma normal (1))

Código de producto	Nombre de producto	Precio unitario
101	MELON	800G
102	Fresa	150G
103	Manzana	120G
104	Limon	200G

Tabla declaración de ventas (segunda forma normal (2))

Código de informe	Código de producto	Cantidad
1101	101	1,100
1101	102	300
1102	103	1,700
1103	104	500
1104	101	2,500
1105	103	2,000
1105	104	700



Tabla (1) contiene los datos relativos a los productos.

Si un valor en la columna de código de producto está determinado, podemos encontrar los valores de otras columnas.



Para los datos relacionados a objetos de declaracion de venta en la tabla (2),

y en esta tabla, la llave primaria determina valores en otras columnas.

Pero...

Para la tabla (2), considerar la combinación de código de reporte y los códigos de producto como llave primaria.

Llave primaria	
Codigo de Informe	Codigo de producto

En algunos casos, dos tipos de productos se venden al mismo tiempo...

En otros casos, un tipo de producto se vende en cantidades diferentes.

Esto significa que...

Se divide la tabla de modo que cuando una clave primaria está determinada, los valores de otras columnas se determinan.

¿Entiendes?

Ya veo.

①

Codigo de producto	Nombre de producto	Precio unitario
--------------------	--------------------	-----------------

Llave primaria

②

Codigo de informe	Codigo de producto	Cantidad
-------------------	--------------------	----------

Llave primaria

La tabla que resulta de la división de acuerdo con esta regla se llama



La segunda forma normal.

Podemos agregar las mandarinas que hablábamos antes para la segunda forma normal (1)

Incluso si el precio del melón cambia, sólo corregimos los datos en una fila, ¿verdad?

También podemos añadir los kiwis y uvas.

¡Lo que no se han vendido todavía!



... Por cierto, se divide la primera forma normal (2),

UH?



¿Así que no es necesario dividir la primera forma normal de tabla venta (1)?

oh, estás usando lentes ahora.

FLOOMP

¡Compruebal

¡Buen punto!



Tabla de ventas
(Primera forma normal (1))

Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación
1101	3/5	12	El Reino de MINAMI
1102	3/7	23	Imperio Alfa
1103	3/8	25	El Reino de RITOL
1104	3/10	12	El Reino de MINAMI
1105	3/12	25	El Reino de RITOL



Para esta tabla, si un valor en los códigos de informes se determina, todos los demás valores de fecha, código de destino de las exportaciones, y el nombre del destino de las exportaciones se determinan.

¡Bien!

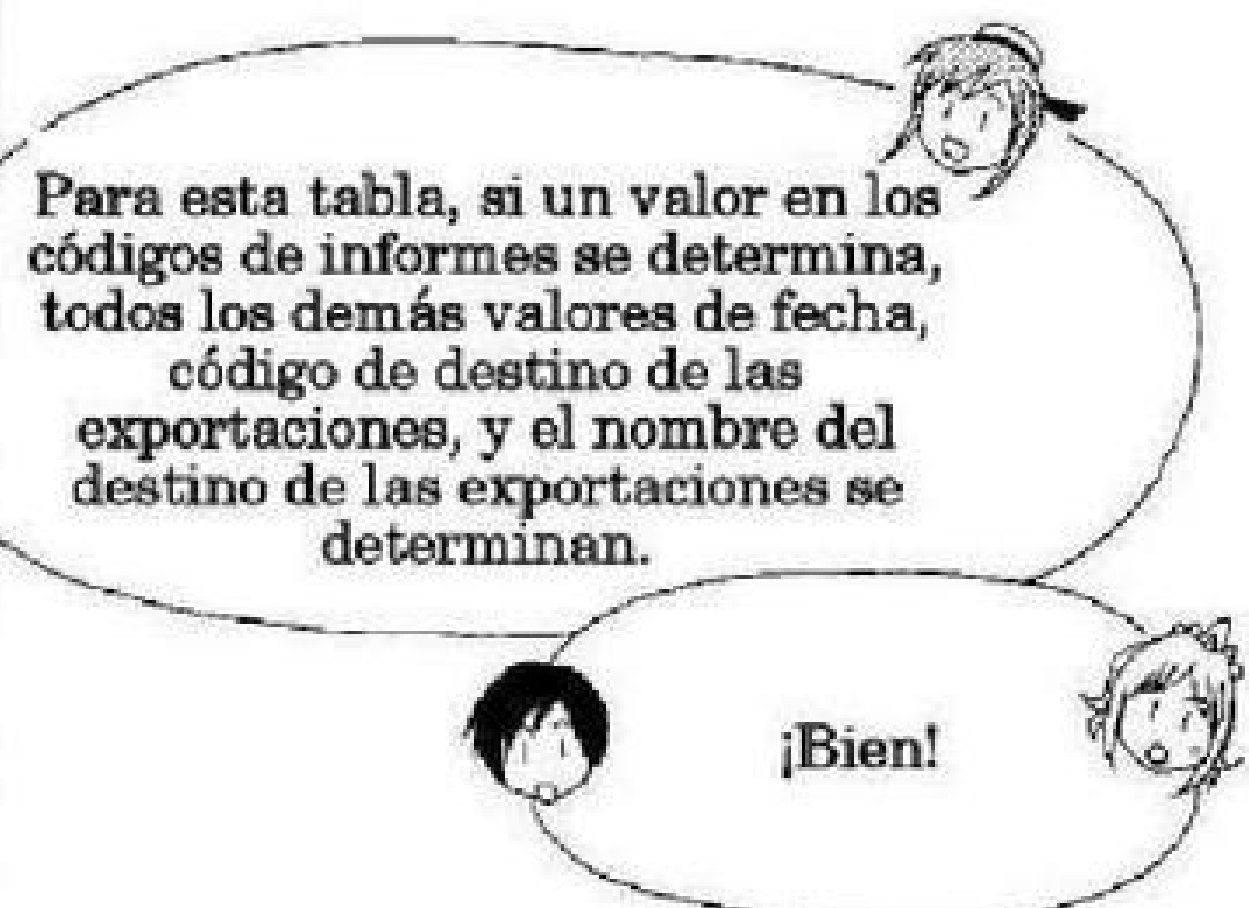




Tabla de ventas (Primera forma normal (1))				Tabla de ventas (Segunda forma normal (3))			
Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación	Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación
Eso es correcto. Puedes considerar la primera forma normal (1)...			El Reino de MINAMI	¡Como la segunda forma normal (3)!			El Reino de MINAMI
			Imperio Alfa				Imperio Alfa
			El Reino de RITOL				El Reino de RITOL



Mira la segunda forma normal (3) de nuevo.

HUH?

SALES TABLE

No es posible administrar los destinos de exportación con esta tabla.

Piensa, piensa, piensa...
¡Ah!

SALES TABLE
(SECOND NORMAL FORM (3))

REPORT CODE	DATE	EXPORT DEST. CODE	EXPORT DEST. NAME
1101	3/5	12	THE KINGDOM OF MINANMI
	3/7	23	ALPHA EMPIRE
	3/8	25	THE KINGDOM OF RITOL

THE KINGDOM OF SAZANNA, TO WHICH NO FRUIT HAS BEEN EXPORTED, CANNOT BE MANAGED BY ADDING IT TO THIS TABLE.



En la tabla (3), los datos relativos a los destinos de exportación y las ventas se mezcla.

La falta de datos



HMM...

¡Una vez más, dividir!

¿Cómo podemos manejar los destinos de exportación de forma independiente?

Así es...

Tabla de ventas
(Tercera forma normal (1))

Codigo de informe	Fecha	Codigo destino de exportacion
1101	3/5	12
1102	3/7	23
1103	3/8	25
1104	3/10	12
1105	3/12	25

Tabla destino de exportacion
(Tercera forma normal (2))

Codigo destino de exportacion	Nombre destino de exportacion
12	El Reino de MINANMI
23	Imperio Alfa
25	El Reino de RITOL

SHAZAM!

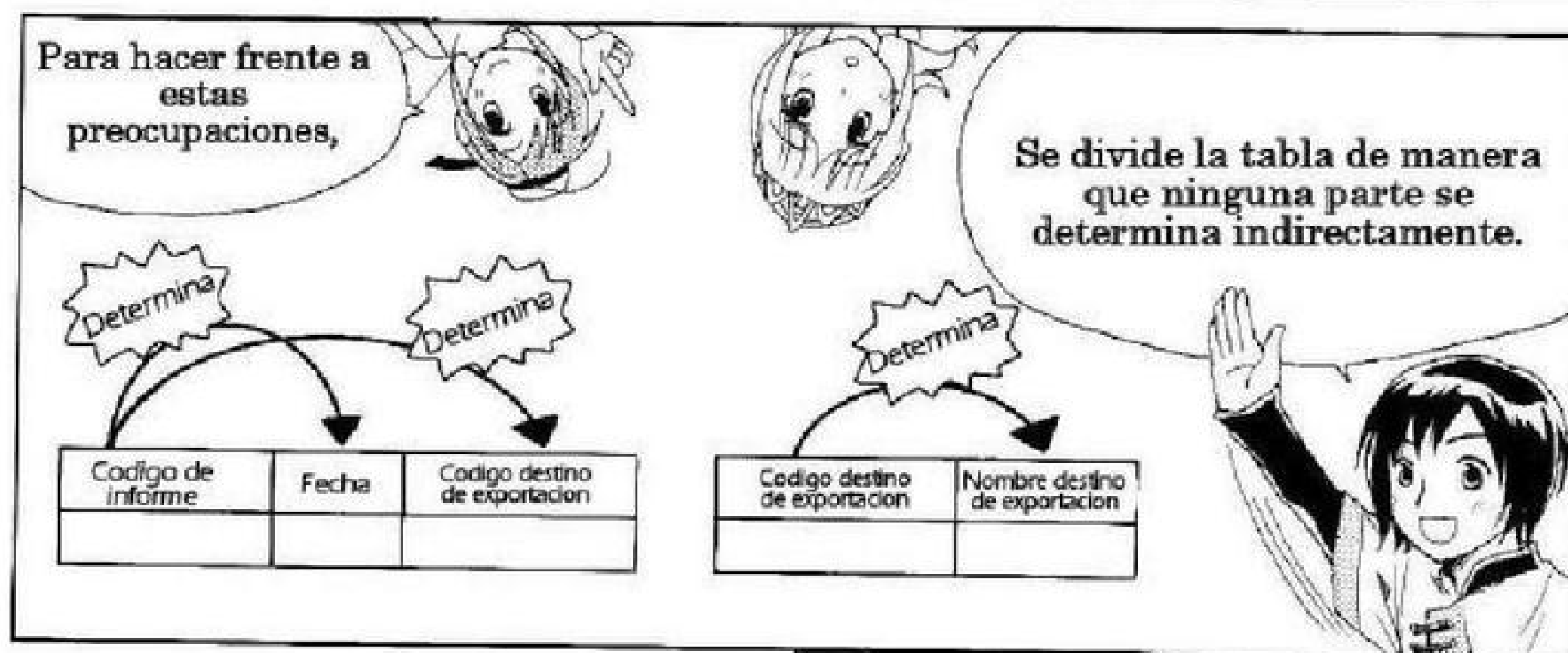
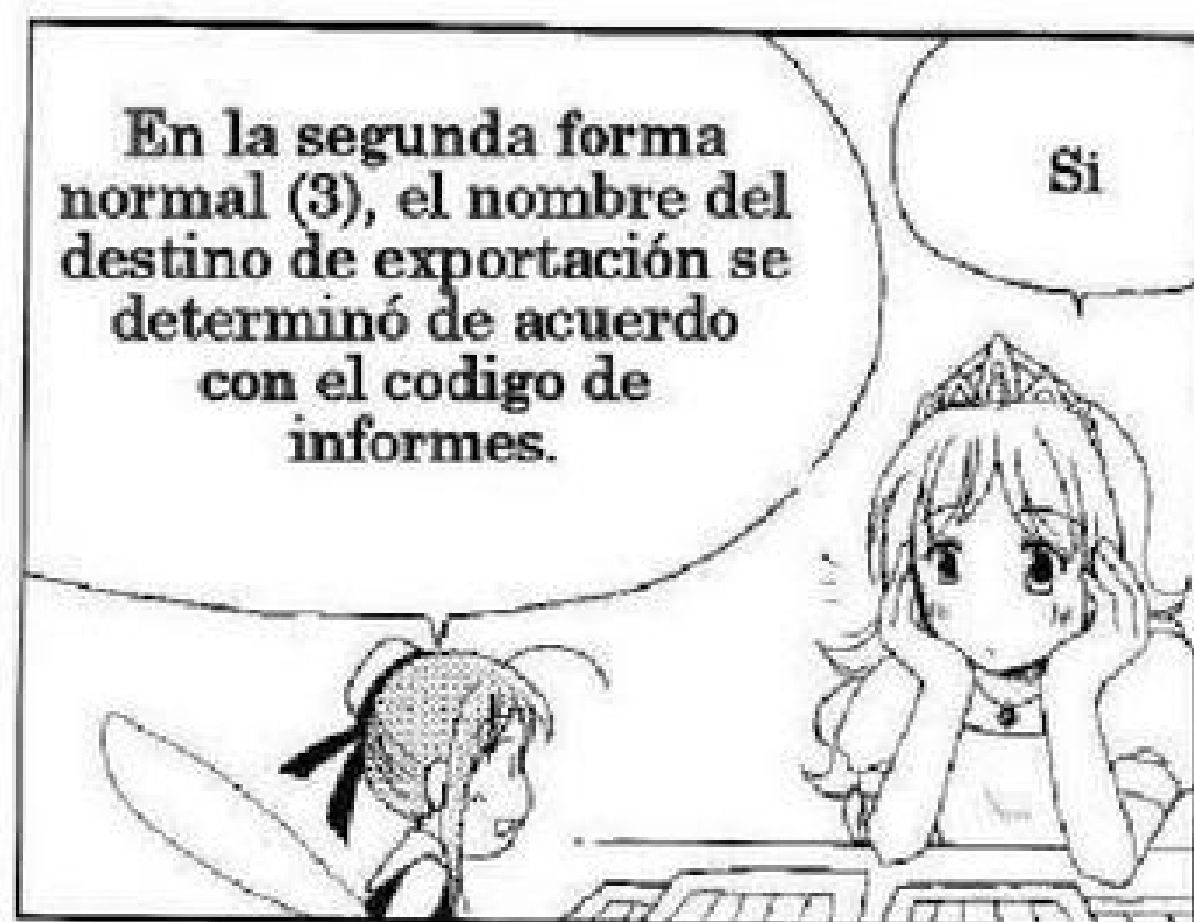


Tabla de venta

Codigo del reporte	DATE	Codigo destino de exportacion
1101	3/5	12
1102	3/7	23
1103	3/8	25
1104	3/10	12
1105	3/12	25

Tabla destino de exportación

Codigo destino de exportacion	Nombre destino de exportacion
12	El Reino de Minanmi
23	Imperio Alfa
25	El Reino de Ritol

Tabla declaración de ventas

Codigo del informe	Codigo de producto	Cantidad
1101	101	1,100
1101	102	300
1102	103	1,700
1103	104	500
1104	101	2,500
1105	103	2,000
1105	104	700

Tabla de productos

Codigo de producto	Nombre de producto	Precio unitario
101	MELON	800G
102	Fresa	150G
103	Manzana	120G
104	Limon	200G

Estos son los cuadros que resultan cuando se divide una tabla hasta la tercera forma normal.

Una base de datos relacional normalmente utiliza las tablas divididas hasta la tercera forma normal.

¡Ahora, nuestra base de datos está completa!

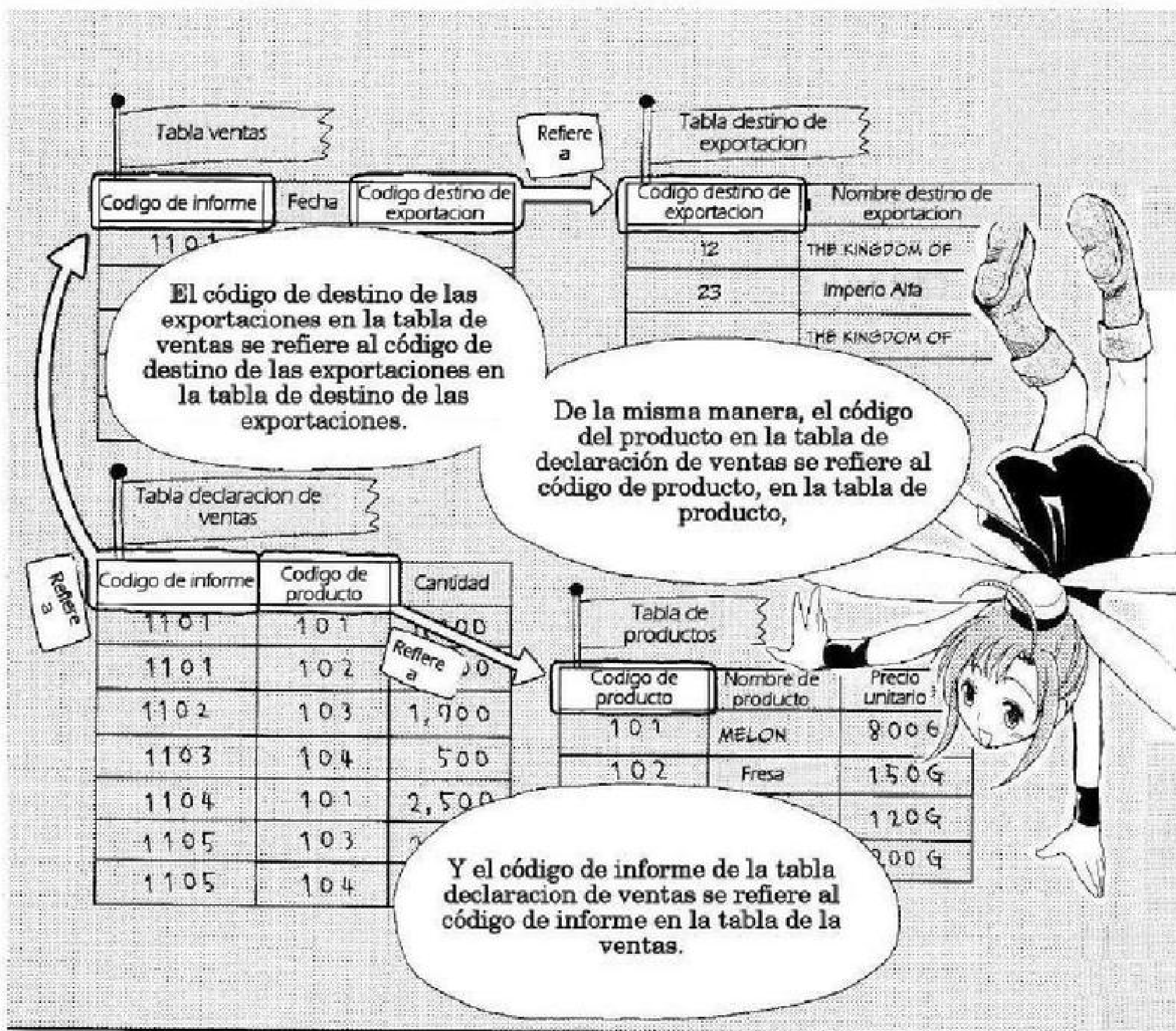
UP HIGH!

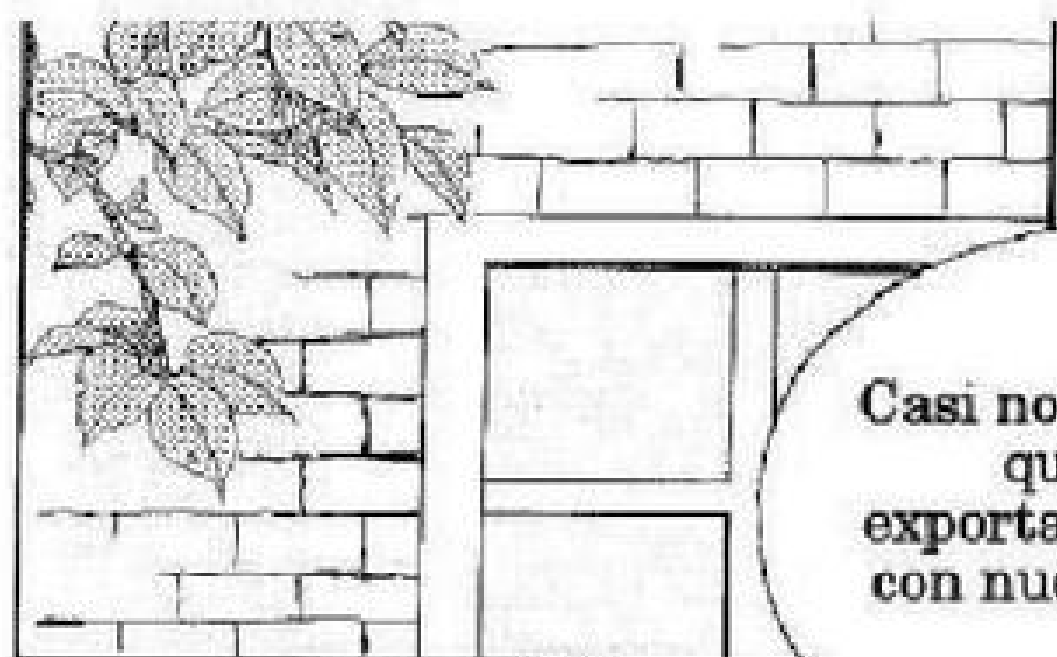
Sobresaltado

CAIN? ? PRINCESS?

~ZOWIE!~ ♡







Casi no puedo esperar para que la gestión de exportación funcione mejor con nuestra base de datos.

¡Estruendo!

Cansado

Si

Oh, eres tú.

¿Ocurre algo?

Princesa Sr. CAIN...

Ustedes dos han estado actuando raro.

Oh, escucha... Puedo explicarlo.

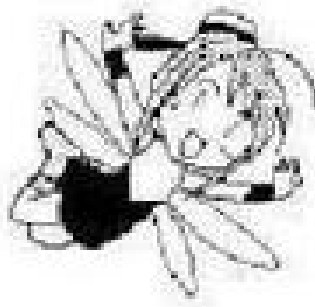
¿Qué le pasa a usted?

Usted no le dijo a la princesa algo extraño, ¿verdad Sr. Cain?

TEE-HEE

¿Yo? Por supuesto que no.

¿QUÉ ES EL MODELO E-R?



Princesa Ruruna y Caín se han dado cuenta de la situación real del Reino de Kod utilizando un modelo E-R (Entidad-Relación). Cuando intenta crear una base de datos por sí mismo, el primer paso es determinar las condiciones de los datos que está intentando modelar.

Usando el modelo E-R, trata de definir una entidad en sus datos. Una entidad es un objeto del mundo real o "cosa", como la fruta o destino de las exportaciones.

Además, un modelo E-R muestra la relación entre las entidades. Princesa Ruruna y Caín realizaron su análisis en el supuesto de que existe una relación entre las llamadas ventas de frutas y destino de las exportaciones. La fruta se exporta a varios destinos de exportación. Aunque cada destino de las exportaciones también importa múltiples tipos de fruta. Por esta razón, se realizó un análisis para el modelo E-R asumiendo que hubo una relación llamada de muchos a muchos entre la fruta y los destinos de exportación. M fruta tiene una relación con destinos de exportación N. El número de asociaciones entre las entidades que se llama cardinalidad.

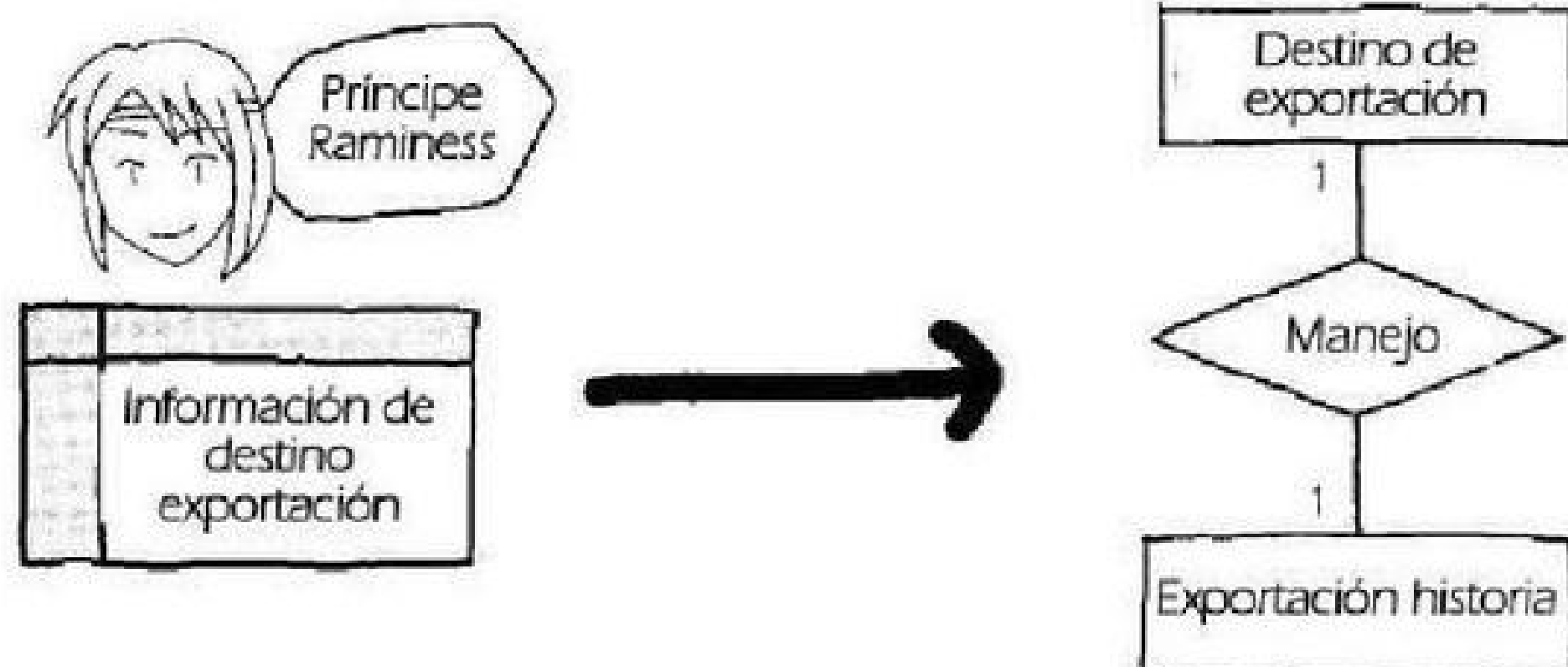


CÓMO ANALIZAR EL MODELO E-R

¿Cómo realizar análisis en los siguientes casos? Piensa en ello.

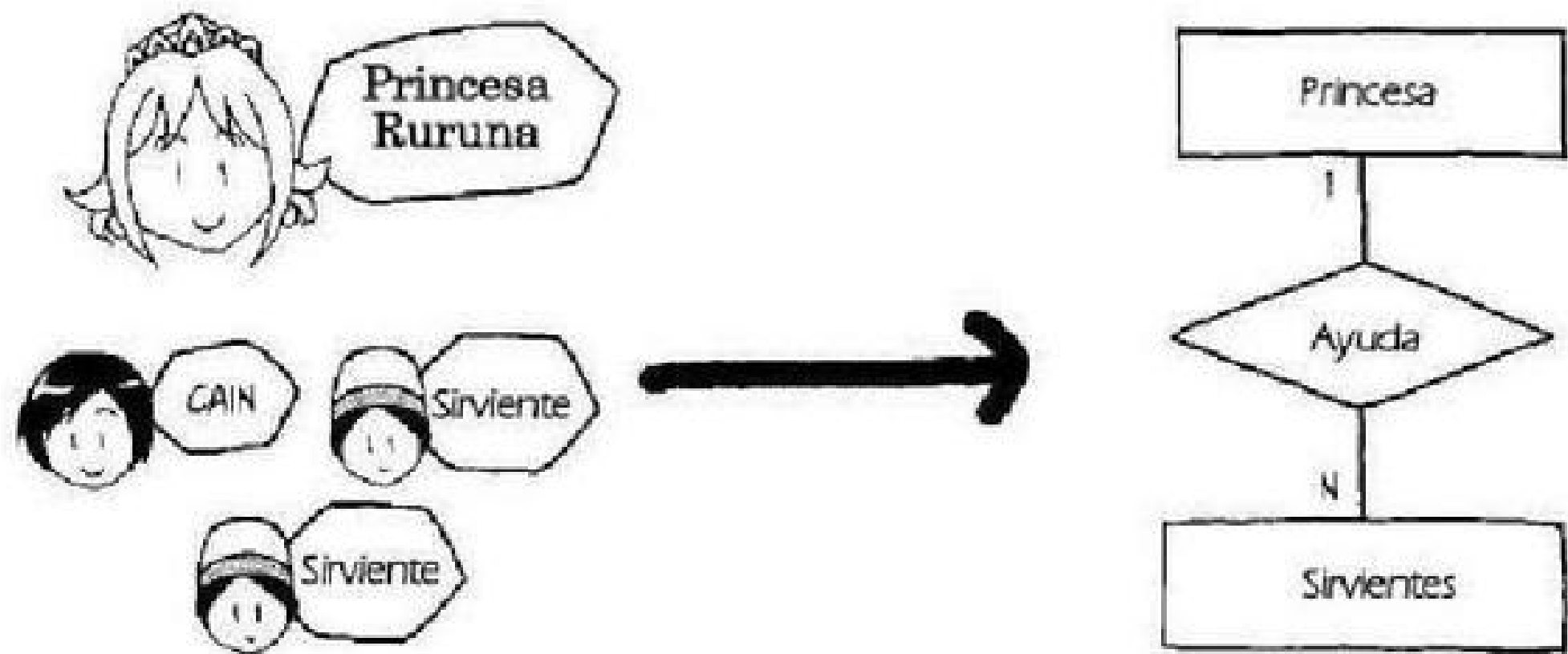
CASO 1: RELACIÓN UNO-A-UNO

Un destino de las exportaciones administra una pieza de información de exportación de la historia. Este tipo de relación se denomina una relación de uno a uno.



CASO 2: RELACION UNO-A-MUCHOS

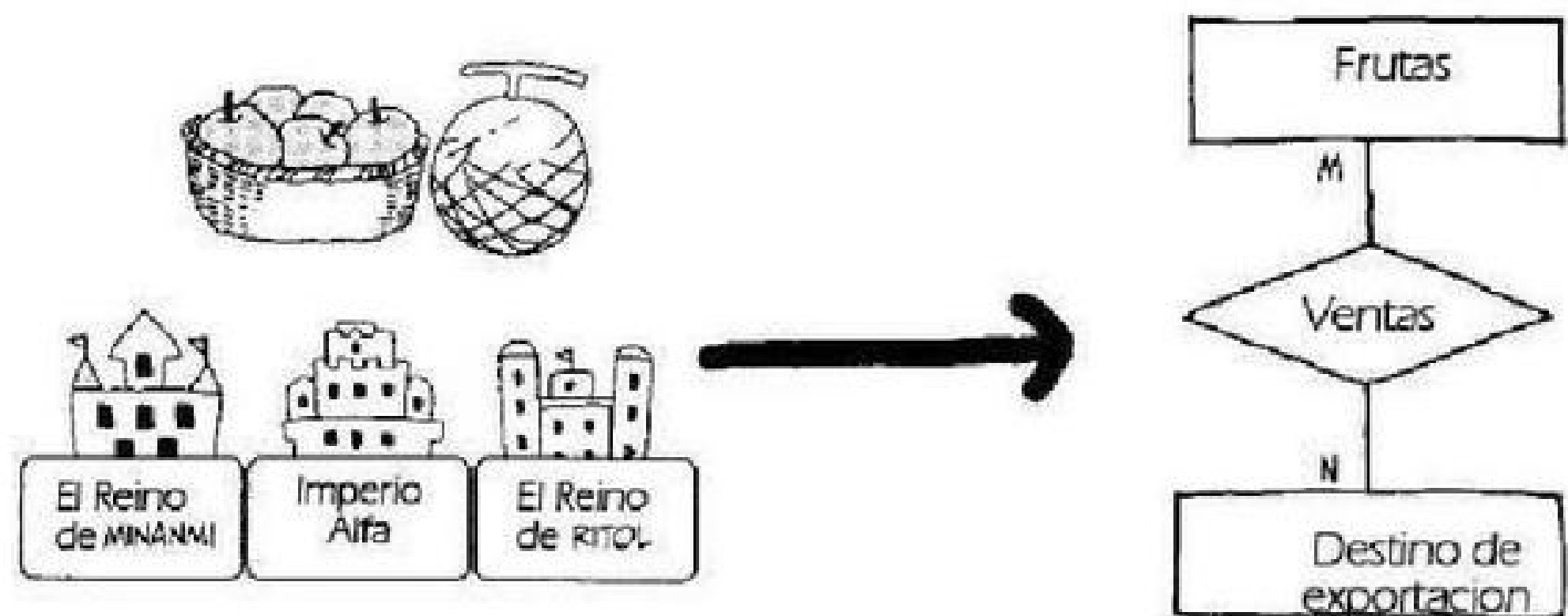
Múltiples sirviente sirven a una princesa. Los sirvientes no pueden servir a ninguna otra princesa o incluso el rey.



Este tipo de relación se denomina una relación de uno a muchos.

CASO 3: MUCHOS-A-MUCHOS

La fruta se exporta a varios destinos de exportación. Los destinos de las exportaciones importa múltiple tipo de fruta.



Este tipo de relación se denomina relación de muchos a muchos.

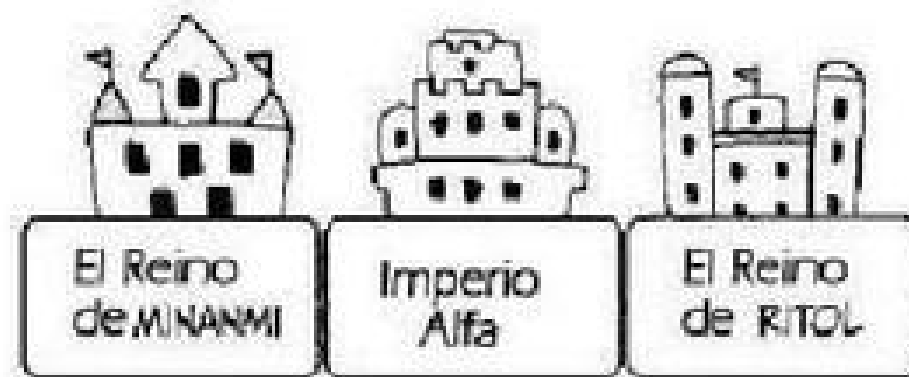


PREGUNTAS

¿Qué tan bien entiende el modelo E-R? Analizar y trazar un modelo ER para cada uno de los casos más adelante. Las respuestas están en la página 82.

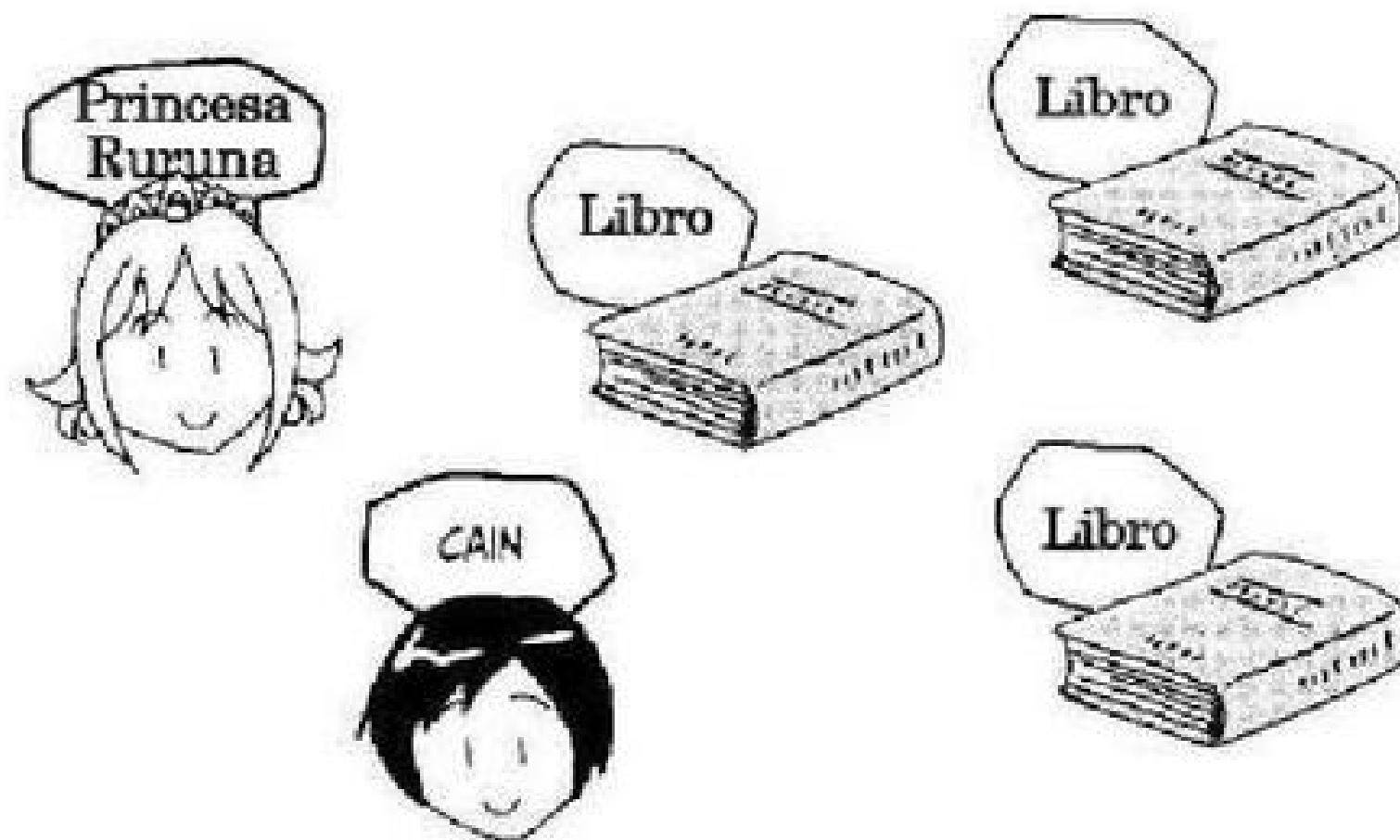
Q1

Un miembro del personal administra varios clientes. Un cliente nunca será contactado por mas de un miembro del personal.

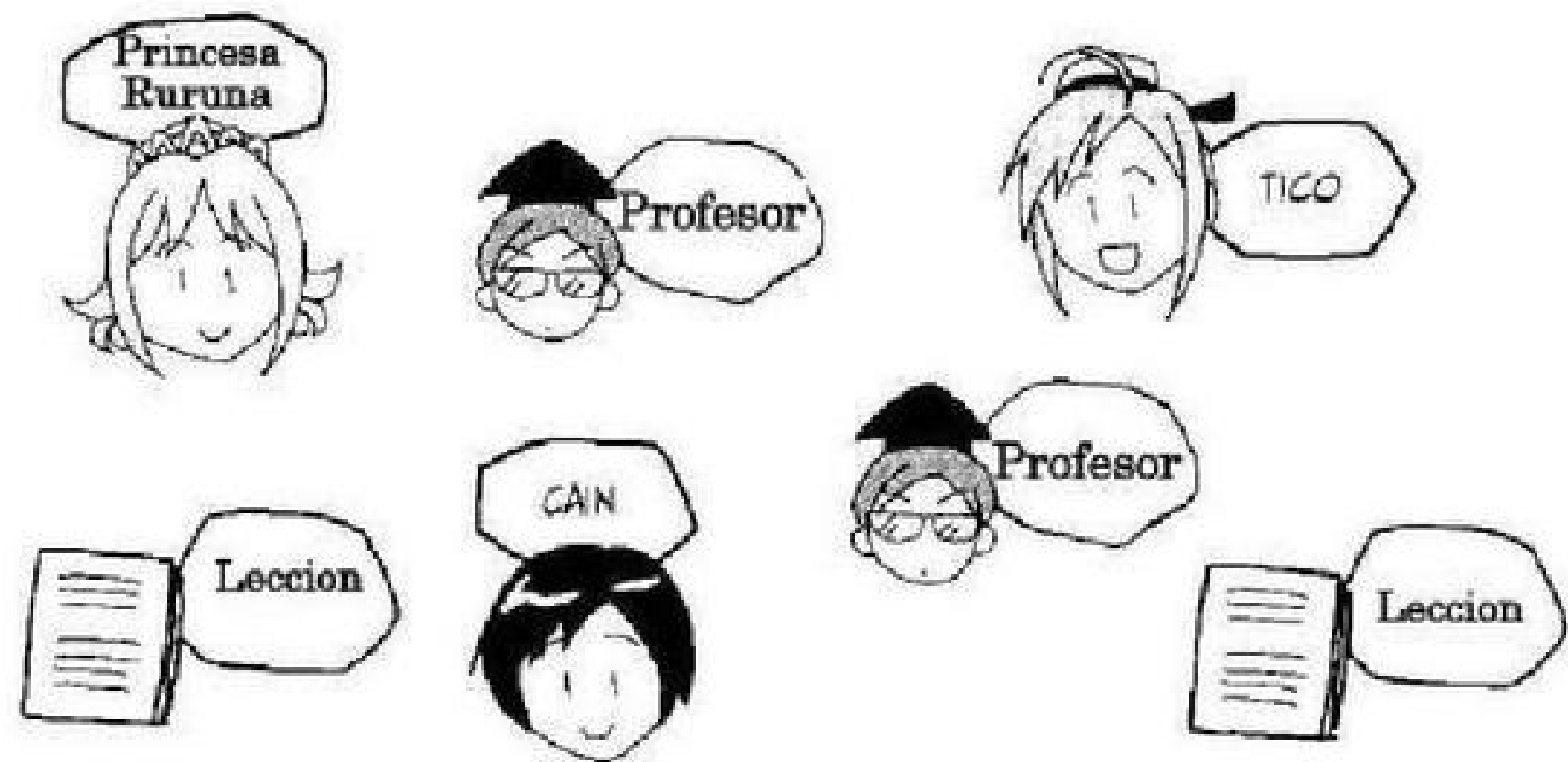


Q2

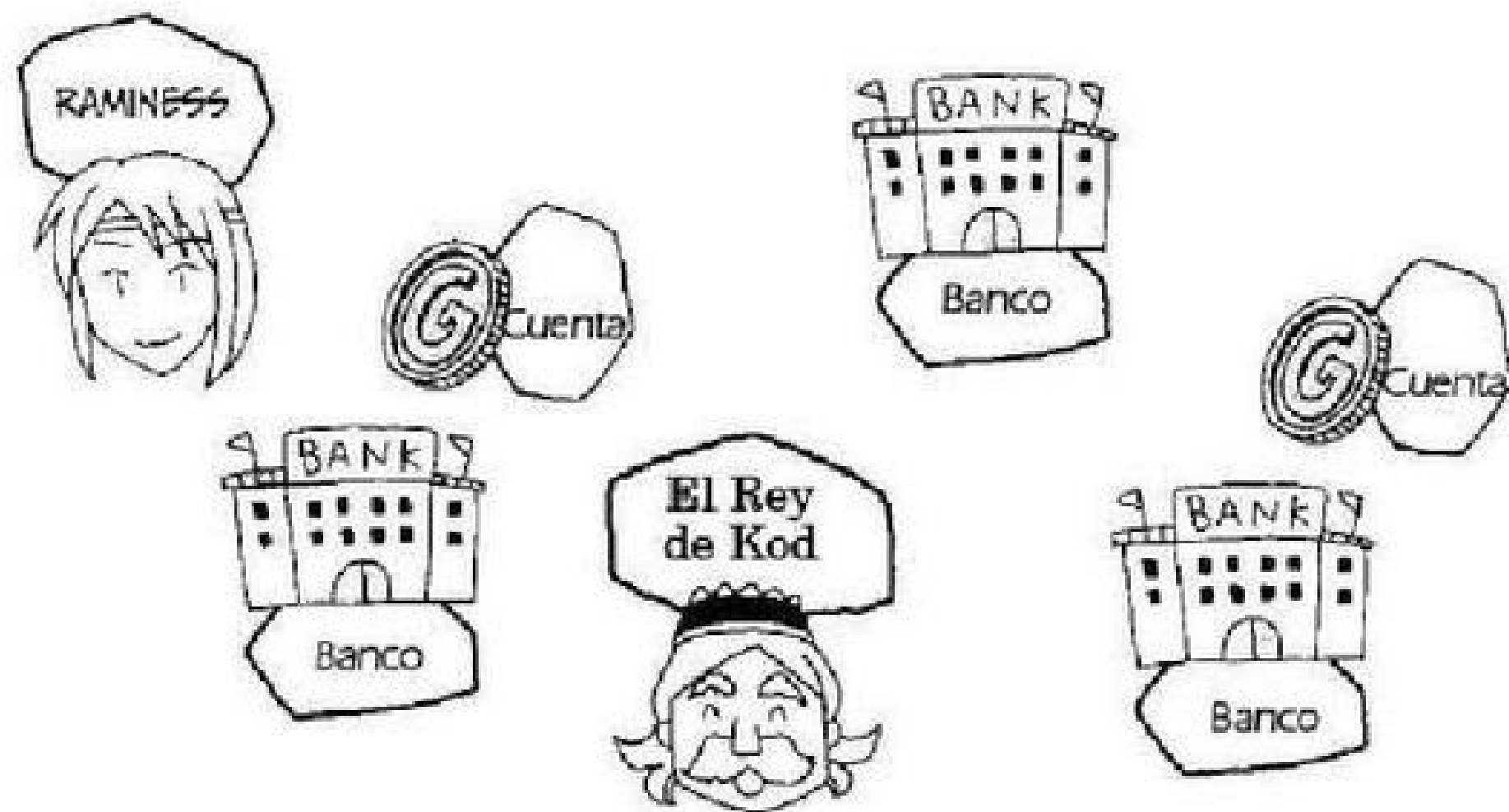
Una persona puede sacar múltiples libros. Los libros pueden ser prestados a varios estudiantes en diferentes momentos.



- Q3 Cada estudiante asiste a clases múltiples. Cada lección es atendida por varios estudiantes. Un profesor da clases múltiples. Cada lección está dada por un profesor.



- Q4 Cada cliente puede abrir varias cuentas de depósito. Cada cuenta de depósito es abierto por un cliente. Cada banco administra varias cuentas de depósito. Cada cuenta de depósito es administrado por un banco.



Tenga en cuenta que E-R analiza basado en modelos no necesariamente produce un resultado 'correcto'. Puede haber muchas maneras de organizar lógicamente datos para reflejar las condiciones reales.

NORMALIZAR UNA TABLA



Princesa Ruruna y Cain aprendió acerca de la normalización, el proceso de tabulación de los datos del mundo real para una base de datos relacional. Es necesario normalizar los datos con el fin de gestionar adecuadamente una base de datos relacional. La normalización se resumen a continuación (los campos sombreados son claves primarias).

Forma no normalizada

Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación	Código de producto	Nombre de producto	Precio unitario	Cantidad
-------------------	-------	-------------------------------	-------------------------------	--------------------	--------------------	-----------------	----------

Primera forma normal

Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación
-------------------	-------	-------------------------------	-------------------------------

Código de informe	Código de producto	Nombre de producto	Precio unitario	Cantidad
-------------------	--------------------	--------------------	-----------------	----------

Segunda forma normal

Código de informe	Fecha	Código destino de exportación	Nombre destino de exportación
-------------------	-------	-------------------------------	-------------------------------

Código de informe	Código de producto	Cantidad
-------------------	--------------------	----------

Código de producto	Nombre de producto	Precio unitario
--------------------	--------------------	-----------------

Tercera forma normal

Código de informe	Fecha	Código destino de exportación
-------------------	-------	-------------------------------

Código destino de exportación	Nombre destino de exportación
-------------------------------	-------------------------------

Código de informe	Código de producto	Cantidad
-------------------	--------------------	----------

Código de informe	Nombre de producto	Precio unitario
-------------------	--------------------	-----------------

La forma no normalizada es una tabla en la que los elementos que aparecen más de una vez no se han eliminado. Hemos visto que no se puede manejar datos y utilizar este tipo de tabla de una base de datos relacional. Por consiguiente, es necesario dividir la tabla.

La primera forma normal se refiere a una simple tabla de dos dimensiones que resulta de la división de la tabla original, sin normalizar. Se puede considerar que sea tabla con un elemento en cada célula. La tabla se divide de manera que no aparecerá ningún elemento más de una vez.

La segunda forma normal se refiere a una tabla en la que una clave que identifican datos determina los valores en otras columnas. Aquí, es la clave primaria que determina los valores de otras columnas.

En una base de datos relacional, un valor se llama funcionalmente dependiente si ese valor determina los valores de otra columna. En la segunda forma normal, la tabla se divide de forma que los valores de otras columnas son funcionalmente dependientes de la llave primaria.

En la tercera forma normal, un cuadro está dividido de modo que un valor no está determinado por cualquier clave no primaria. En una base de datos relacional, un valor que se llama transitivamente dependiente si dicho valor determina los valores de otras columnas indirectamente, que es parte de la operación depende funcionalmente. En la tercera forma normal, la tabla está dividida de modo que los valores transitivamente dependientes se eliminan.



PREGUNTAS

Es importante ser capaz de diseñar una tabla de base de datos relacional para diversas situaciones, así que vamos a ver algunos ejemplos de tablas de normalización. Determine cómo la tabla se normalizó en cada uno de los casos más adelante. Las respuestas están en la página 82.

Q5

La siguiente tabla gestiona el préstamo de libros como el ejemplo en la Q2.
¿En qué etapa se encuentra normalizada?

Código de préstamo	Fecha	Código de estudiante	Nombre del estudiante	Dirección del estudiante	Departamento	Año de entrada
--------------------	-------	----------------------	-----------------------	--------------------------	--------------	----------------

ISBN	Nombre del libro	Nombre del autor	Fecha de publicación	Número total de páginas
------	------------------	------------------	----------------------	-------------------------

Código de préstamo	ISBN	Cantidad
--------------------	------	----------

Q6

La siguiente tabla muestra también una situación de préstamo de libros. ¿En qué etapa se encuentra normalizada?

Código de préstamo	Fecha	Código de estudiante
--------------------	-------	----------------------

Código de estudiante	Nombre del estudiante	Dirección del estudiante	Departamento	Año de entrada
----------------------	-----------------------	--------------------------	--------------	----------------

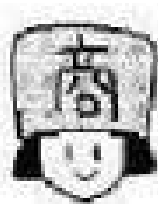
ISBN	Nombre del libro	Nombre del autor	Fecha de publicación	Número total de páginas
------	------------------	------------------	----------------------	-------------------------

Código de préstamo	ISBN	Cantidad
--------------------	------	----------

Q7

La siguiente tabla muestra las ventas mensuales de cada miembro del personal. Cada departamento cuenta con miembros del personal múltiple. Un miembro del personal sólo puede ser parte de un departamento. Normalizar esta tabla a la tercera forma normal.

Código de miembro staff	Nombre de miembro staff	Mes	Miembros de ventas	Código de departamento	Nombre de departamento
-------------------------	-------------------------	-----	--------------------	------------------------	------------------------



Departamento de mercancías



Departamento de regalos en el extranjero



Departamento de exportación



Serviente



Serviente



Serviente



Serviente



Serviente

Q8

La siguiente tabla representa un sistema de orden de recepción. Normalizar a la tercera forma normal. Sin embargo, tratar a un cliente según el código 'toma de pedidos'. Puedes procesar varios productos basados en un código 'toma de pedidos'. Además, un código 'toma de pedidos' debe corresponder a un sólo representante.

Código de 'Toma de pedido'	Fecha	Código del cliente	Nombre del cliente	Código de producto	Nombre de producto	Precio unitario	Código representativo	Nombre representativo	Cantidad
----------------------------	-------	--------------------	--------------------	--------------------	--------------------	-----------------	-----------------------	-----------------------	----------

09

La siguiente tabla representa un sistema de orden de recepción. Normalizar a la tercera forma normal. Supongamos que los productos son clasificados por código de producto.

Código de 'Toma de pedido'	Fecha	Código del cliente	Nombre del cliente	Código de producto	Nombre de producto	Precio unitario	Código clasificación de producto	Nombre clasificación de producto	Cantidad
----------------------------	-------	--------------------	--------------------	--------------------	--------------------	-----------------	----------------------------------	----------------------------------	----------

PASO PARA EL DISEÑO DE UNA BASE DE DATOS

¡Has aprendido cómo diseñar una base de datos! Sin embargo, tienes que hacer algo más que eso. Es necesario diseñar una estructura de archivos dentro de la base de datos detallada y diseñar métodos para importar y exportar datos. En general, se puede dividir el diseño de base de datos completa en tres partes: el esquema conceptual, esquema interno y esquema externo.

El esquema conceptual se refiere a un método que modela el mundo real. Es decir, es una manera de determinar la estructura lógica de una base de datos. El esquema conceptual se ha diseñado teniendo en cuenta un modelo E-R basada en la comprensión del mundo actual y la normalización de una tabla.

El esquema interno se refiere a una base de datos visto desde el interior de un ordenador. Es decir, es una manera de determinar la estructura física de una base de datos. El esquema interno está diseñado después de la creación de un método para buscar en la base de datos a alta velocidad.

El esquema externo se refiere a como es vista una base de datos por el usuario o las aplicaciones. El esquema externo está diseñado después de crear los datos necesarios para los programas de aplicación.



Esquema interno

Esquema conceptual

Esquema externo

Princesa Ruruna y Cain ha diseñado una base de datos con un enfoque en el esquema conceptual de este capítulo. Se encuentran en el medio de la mejora de la base de datos.

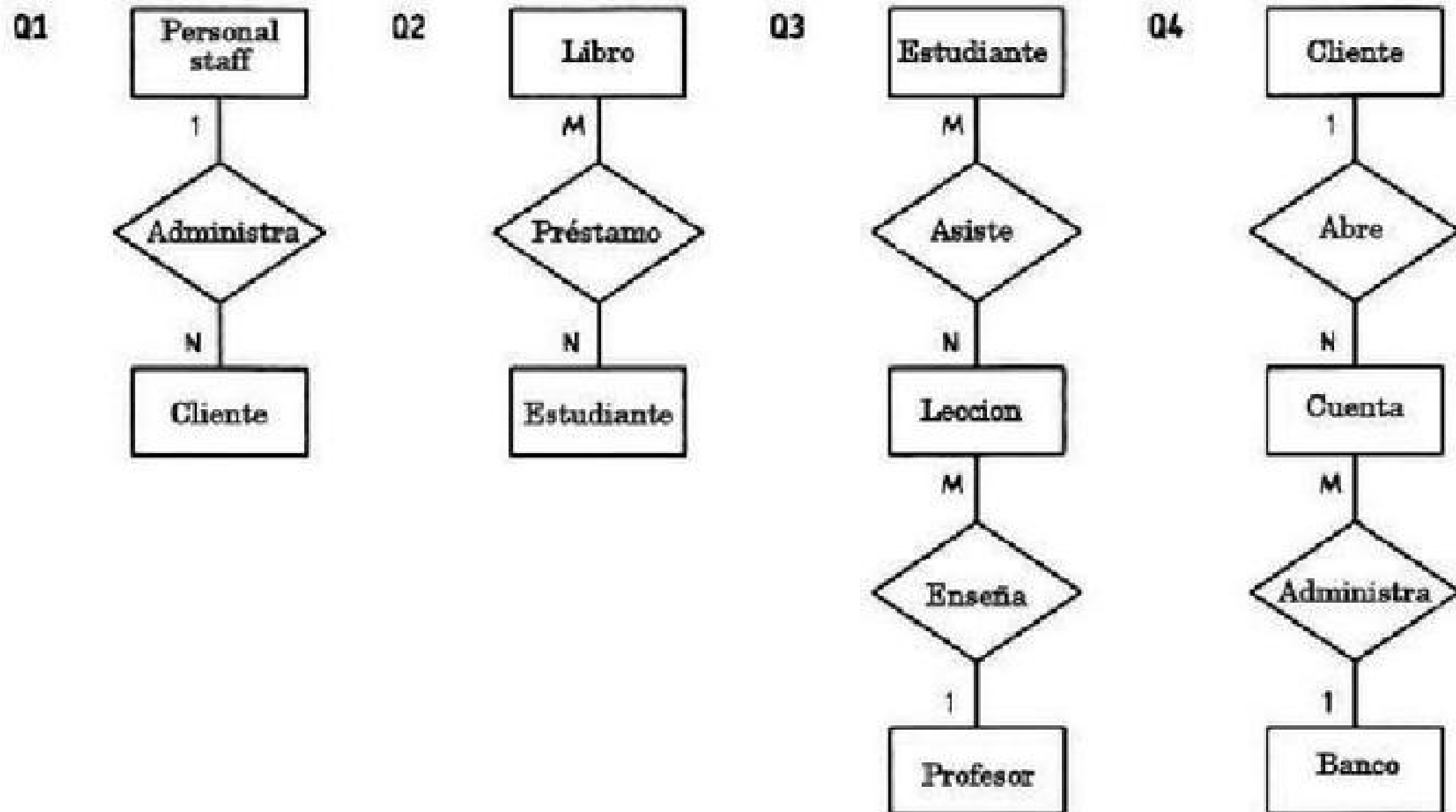
Ahora que ha completado el diseño básico de una base de datos, vamos a ir directamente a la utilización de las bases de datos en el siguiente capítulo.

RESUMEN



- Un modelo E-R se utiliza para analizar las entidades y relaciones.
- Las relaciones entre las entidades pueden ser de uno a uno, uno a muchos y muchos a muchos.
- Los datos de una tabla deben ser normalizados antes de poder utilizarlo para crear una base de datos relacional.
- El diseño de la base de datos se pueden dividir en tres tipos: esquemas conceptuales, esquema interno y esquemas externos.

Respuestas



Q5 Segunda forma normal

Q6 Tercera forma normal

Q7

Codigo miembro staff	Mes	Miembros de ventas
----------------------	-----	--------------------

Codigo miembro staff	Nombre miembro staff	Codigo de departamento
----------------------	----------------------	------------------------

Codigo de departamento	Nombre de departamento
------------------------	------------------------

Q8

Código de 'Toma de pedido'	Fecha	Código del cliente	Código representativo
----------------------------	-------	--------------------	-----------------------

Código del cliente	Nombre del cliente
--------------------	--------------------

Código de 'Toma de pedido'	Código de producto	Cantidad
----------------------------	--------------------	----------

Código de producto	Nombre de producto	Precio unitario
--------------------	--------------------	-----------------

Código representativo	Nombre representativo
-----------------------	-----------------------

Q9

Código de 'Toma de pedido'	Fecha	Código del cliente
----------------------------	-------	--------------------

Código del cliente	Nombre del cliente
--------------------	--------------------

Código de 'Toma de pedido'	Código de producto	Cantidad
----------------------------	--------------------	----------

Código de producto	Código clasificación de producto	Nombre de producto	Precio unitario
--------------------	----------------------------------	--------------------	-----------------

Código clasificación de producto	Nombre clasificación de producto
----------------------------------	----------------------------------

DISEÑO DE UNA BASE DE DATOS

En este capítulo, el aprendió a diseñar una base de datos relacional. Sin embargo, existen otros métodos de diseño de la base de datos. Usabilidad y la eficacia de una base de datos depende de un análisis y método de diseño. Por lo tanto, es importante para crear una base de datos apropiada en la etapa de diseño.

En la etapa de diseño de base de datos, es necesario realizar diversas tareas además de diseño de la tabla. Por ejemplo, debe tener en cuenta un tipo de datos a utilizar en la tabla. También puede ser necesario para especificar las columnas que indican valores numéricos, Monedas y cadenas de caracteres. Además, es necesario idear un método de búsqueda para que pueda llevar a cabo búsquedas rápidas. A veces, es necesario crear un diseño, manteniendo la organización de archivos físicos en mente. Y tienes que controlar que usuarios pueden acceder a la base de datos para garantizar la seguridad. Hay muchos factores que usted necesita considerar cuando se diseña una base de datos. Veremos algunos de estos factores en los capítulos siguientes.

4

¡VAMOS A APRENDER SOBRE SQL!



USO DE SQL

Caminando por el pueblo me hace recordar mi infancia.

¡Ja, ja, ja!

A menudo se saltaba las clases y se escabulló del castillo.

Hace muchos años...

!Princesa!
princesa
!Ruruna!

CLATTER

¿No?

¡No se puede salir del castillo cuando quiera!

HUFF
HUFF
HUFF







¿Qué te pasa?...

OH, NO, NO...

¡¡nada!!

¿Qué pasa conmigo?...

¿Qué?



!!

HA, HA, HA



Príncipe
Raminess

ZOWIE!!

Vecinos,
♥ vecinos

HA, HA, HA

¡¡Qué
bambón!!

RAMINESS!!

¿Cómo están
bellas damas?



Oh no, él viene
hacia aquí.

¡Caín! Vamos a ir
en esta cafetería.

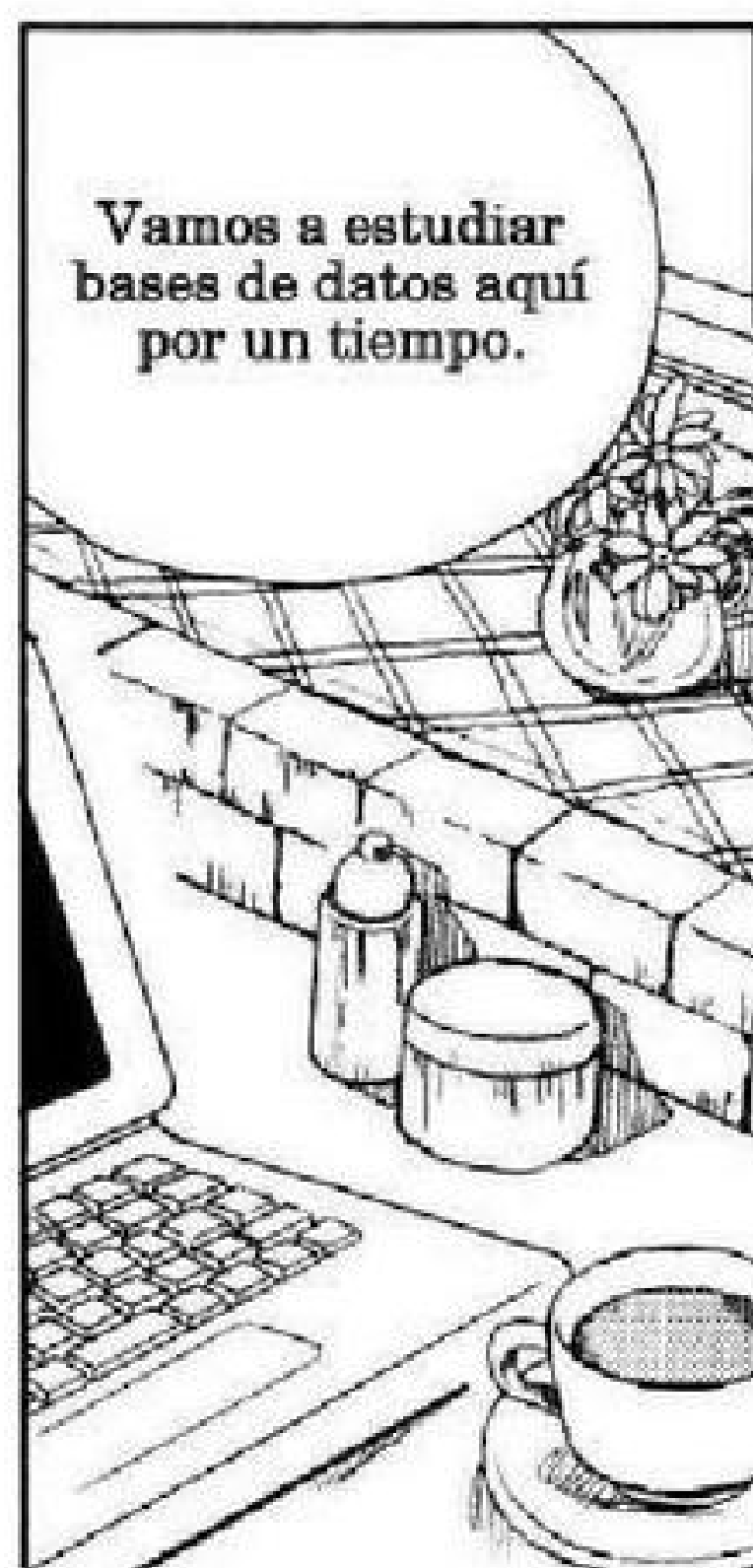
¿Qué?

SQUEAL!
EEK!

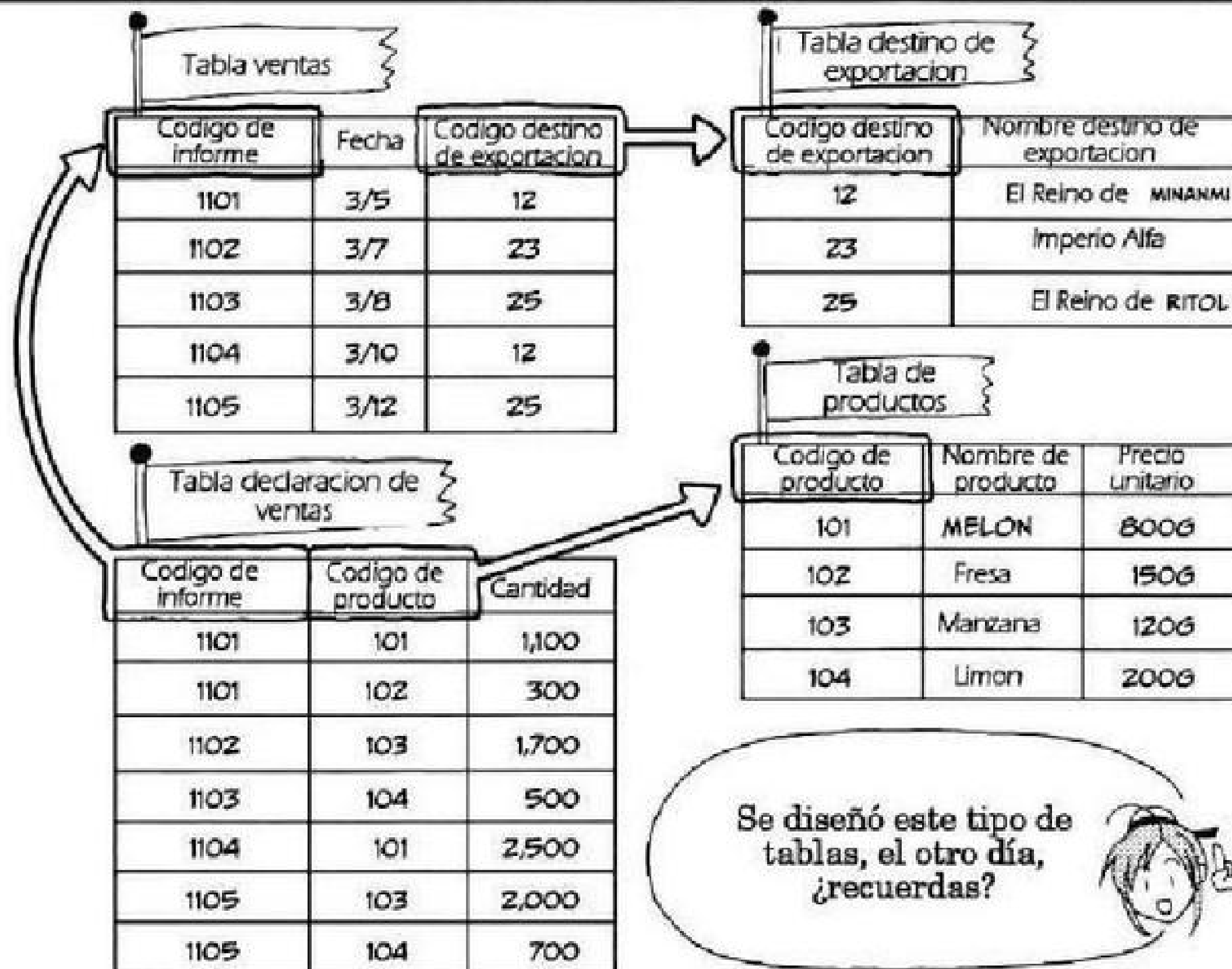
HA, HA, HA



Vamos a estudiar
bases de datos aquí
por un tiempo.









SEARCHING FOR DATA USING a *SELECT* STATEMENT



Estos grupos de palabras se llaman frases.

En SQL, puede especificar un nombre de columna que desea recuperar con la frase y seleccionas el nombre de la tabla de la que desea recuperar de la frase.

FROM

Tabla de productos

Código de producto	Nombre de producto	Precio unitario
101	MELON	800G
102	Fresa	150G
103	Manzana	120G
104	Limon	200G

SELECT

Estos son los datos recuperados.

Esto le permite recuperar todos los nombres de producto de la tabla de productos.

¡Aquí están! ♪

Nombre de producto

MELON

Fresa

Manzana

Limon

Estamos teniendo una conversación con una base de datos utilizando SQL.

Eso es correcto. Puede recuperar los datos necesarios mediante el uso de diversos tipos de frases.

Varios tipos... hmm.

Pues bien, por ejemplo,

¿Acerca de cómo solicitar una lista de productos cuyo precio unitario es mayor o igual a 200G?

Mayor o igual que 200G





Entonces, ¿qué hacer para recuperar la manzana, por ejemplo?

Escríbelo como este. Cuando usas caracteres como una condición, encerrarlos dentro de comillas (').

```
SELECT *  
FROM producto  
WHERE nombre_producto = 'manzana';
```

Código de	Nombre de producto	Precio unitario
103	manzana	120G





Expresar la parte desconocida usando %, así...

Esto recuperará los nombres de productos que terminan con n.

MELON Limon

```
SELECT *  
FROM producto  
WHERE nombre_producto LIKE '%n';
```

Codigo de producto	Nombre de producto	Precio unitario
101	MELON	800G
104	Limon	200G



USING AGGREGATE FUNCTIONS

También puede ordenar los resultados obtenidos con una orden por frase.

Para ordenar los productos por orden de precio ascendente, agregar una declaración como ordenar por precio unitario.

Usted puede encontrar información acerca de los productos al hacer esto.

¡Eso es genial!

```
SELECT *  
FROM producto  
WHERE nombre_producto LIKE '%n';  
ORDER BY precio_unitario;
```

Código de producto	Nombre de producto	Precio unitario
103	Manzana	1206
102	Fresa	1506

¡Quiero saber más sobre SQL, Tico!

¿En serio?

Me alegro.

¿Qué tal este?

En la frase seleccionar, usar AVG (nombre de la columna) para obtener el promedio de cada fila.

```
SELECT AVG (precio_unitario)  
FROM producto ;
```

はっ!!

Es increíble.

ZZOOP!!

Precio unitario promedio
3.175

Ahora tenemos el precio unitario promedio de los productos.



Las funciones de agregación de SQL

Función	Descripción
COUNT (*)	Obtiene el número de fila
COUNT (columna_nombre)	Obtiene el número de veces que la columna no es nulo
COUNT (DISTINCT columna_nombre)	Obtiene el número de valores distintos en la columna
SUM (columna_nombre)	Obtiene la suma de los valores de la columna en todas las filas
AVG (columna_nombre)	Obtiene el promedio de los valores de la columna en todas las filas
MAX (columna_nombre)	Obtiene el valor máximo de la columna
MIN (columna_nombre)	Obtiene el valor mínimo de la columna





Combinar tablas



Combinar tablas mediante la colocación de una coma entre ellos.

Si el mismo nombre de columna aparece en varias tablas, simplemente especifique como Table_nombre.nombre_columna.

```
SELECT ventas.codigo_informe, fecha, ventas.codigo_exportacion_destino,
nombre_exportacion_destino,
declaracion_ventas.codigo_producto, nombre_producto, precio_unitario,
exportacion_destino

FROM ventas, declaracion_ventas, producto, exportacion_destino

WHERE ventas.codigo_informe = declaracion_ventas.codigo_informe
AND
declaracion_ventas.codigo_producto = producto.codigo_producto
AND
exportacion_destino.codigo_exportacion_destino =
ventas.codigo_exportacion_destino
```

Habiéndose unido a estas cuatro tablas, entonces restringir nuestros resultados utilizando dónde.



De esta manera, puede recuperar datos de informes de ventas de las tablas, aunque estén divididos.

Codigo de Informe	Fecha	Codigo destino de exportacion	Nombre destino de exportacion	Codigo de producto	Nombre de producto	Precio Unitario	Cantidad
1101	3/5	12	El Reino de MINANMI	101	MELON	8006	1,100
1101	3/5	12	El Reino de MINANMI	102	Fresas	1506	300
1102	3/7	23	Imperio Alfa	103	Manzana	1206	1,700
1103	3/8	25	El Reino de RITOL	104	Limon	2006	500
1104	3/10	12	El Reino de MINANMI	101	MELON	8006	2,500
1105	3/12	25	El Reino de RITOL	103	Manzana	1206	2,000
1105	3/12	25	El Reino de RITOL	104	Limon	2006	700

¡Esta igual que la tabla que hemos estado utilizando lo vuelve a crear!

Puede recuperar los datos relativos a las ventas reportar incluso si usted maneja productos, destinos de las exportaciones y las ventas de forma independiente.

¡Eso es genial!

WOW!



Creando tablas



```
CREATE TABLE producto
(
Codigo_producto int NOT NULL,
Nombre_producto varchar (255),
Precio_unitario int,
PRIMARY KEY(Codigo_producto)
);
```

➔

Código de producto	Nombre de producto	Precio Unitario

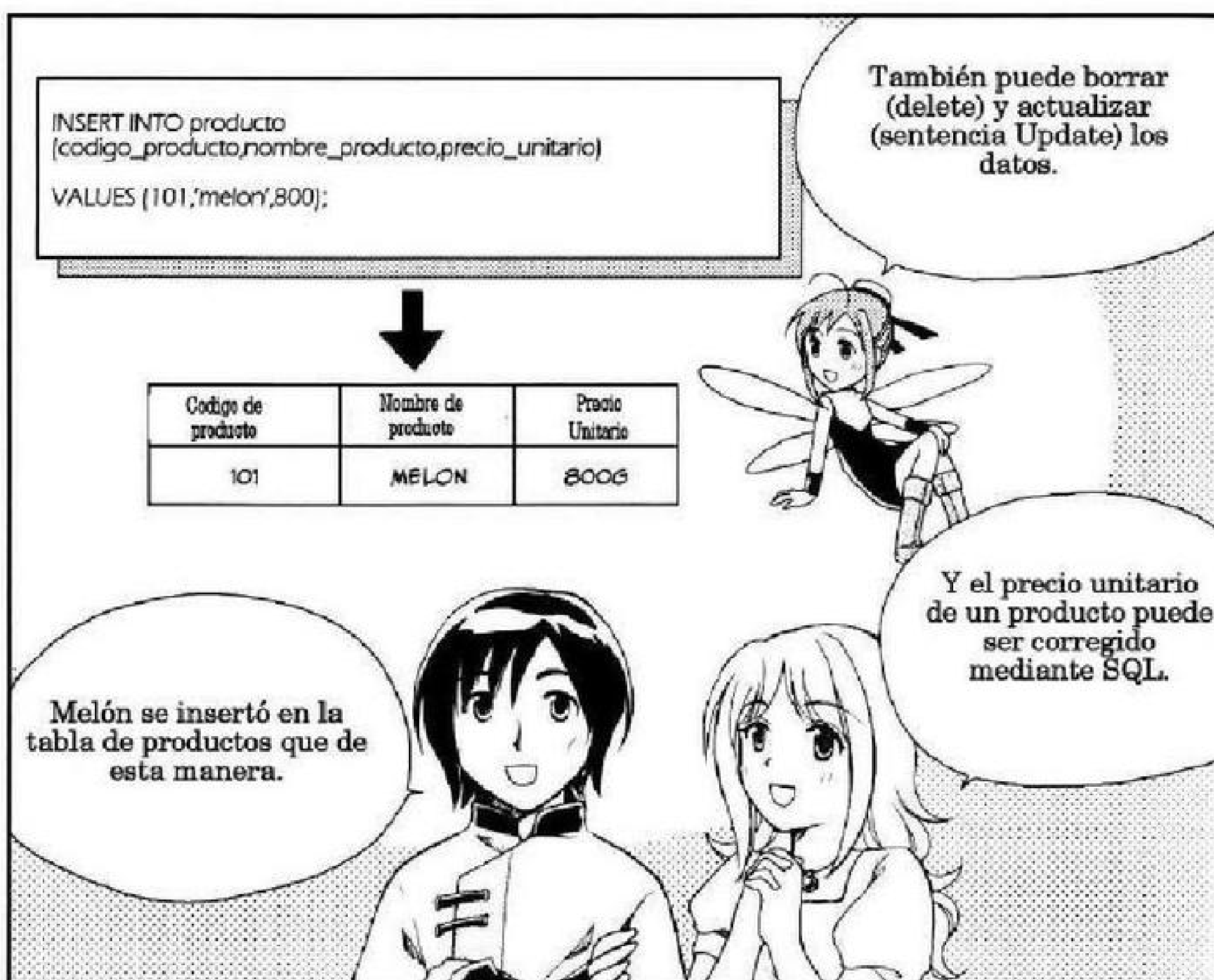
Debe especificar la llave primaria, también. He utilizado el código de producto como llave primaria.

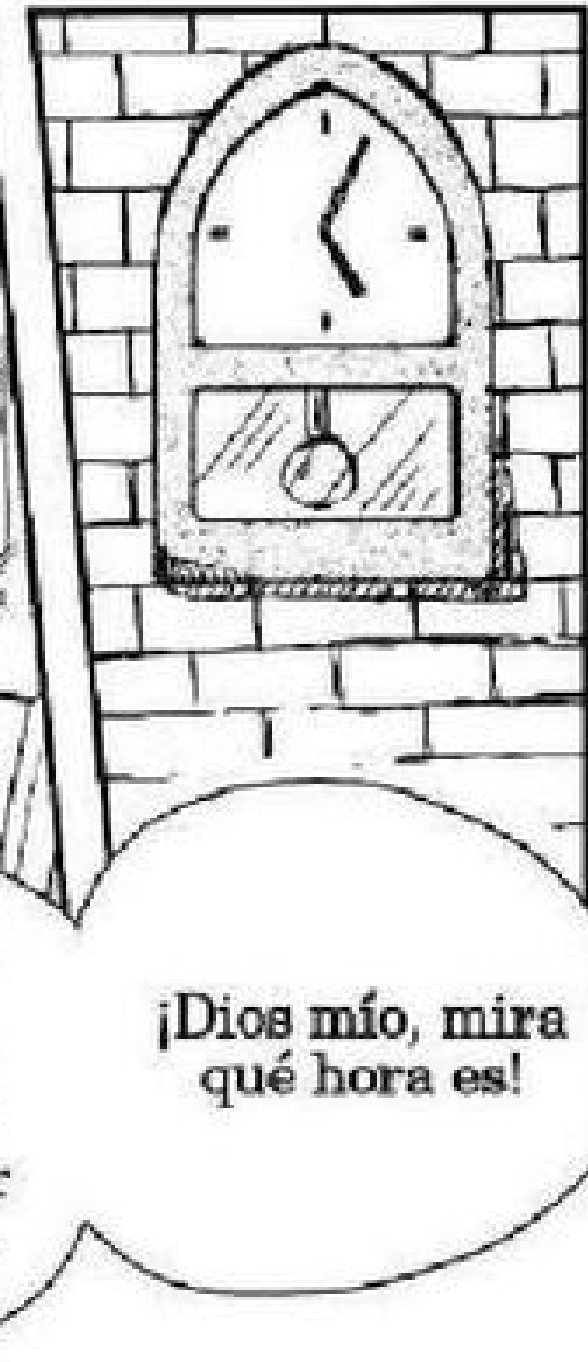
También hemos establecido el tipo de datos de cada columna. Se puede ver que los productos y precio_unitario son enteros (INT). varchar significa que la base de datos espera texto, y (255) limita la nombre_producto a 255 caracteres.

Así...

Esto le impide entrar valores incorrectos.

* SEE PAGE 115 FOR A COMPLETE EXPLANATION OF CREATE TABLE STATEMENTS.





INFORMACIÓN GENERAL DE SQL



En este capítulo, la princesa Ruruna y Cain aprendió acerca de SQL. O la estructura del lenguaje de consulta, un lenguaje utilizado para operar una base de datos relacional. Comandos SQL se pueden dividir en tres tipos distintos:

Lenguaje definición de dato (DDL) Crea una tabla

Lenguaje de manipulación de datos (DML) de entrada y recupera datos

Lenguaje de control de datos (DCL) administra el acceso de usuarios

SQL tiene comandos que crean el marco de una base de datos y un comando que crea una tabla dentro de una base de datos. Puede utilizar este idioma para cambiar y eliminar una tabla también. El lenguaje de base de datos que tiene estas funciones se llama el lenguaje de definición de datos (DDL).

SQL también tiene comandos que manipulan los datos en una base de datos, tales como la inserción, eliminación y actualización de datos. También tiene un comando que le permite buscar datos. El lenguaje de base de datos con estas funciones se llama el lenguaje de manipulación de datos (DML).

Además, SQL ofrece la capacidad para controlar una base de datos, de modo que los conflictos de datos no se producirán incluso si varias personas utilizan la base de datos al mismo tiempo. El lenguaje de base de datos asociada con estas funciones se llama el lenguaje de control de datos (DCL).

LA BÚSQUEDA DE DATOS UTILIZANDO UNA SENTENCIA SELECT

Princesa Ruruna y Cain comenzó a aprender SQL mediante una función básica de datos de búsqueda. Búsquedas SQL para datos cuando una instrucción (una combinación de frases) es la entrada. Para buscar un determinado producto con un precio unitario de 200 g, por ejemplo, se utilizaría la siguiente instrucción SQL.

```
SELECT *  
FROM producto  
WHERE precio_unitario=200
```

Cree una sentencia SQL
mediante la combinación de
frases.

Una instrucción **SELECT** es la instrucción SQL más básica. En él se especifica que la columna, de qué tabla (**FROM**). Y coincidente que las condiciones (**WHERE**). Puede combinar estas frases para hacer intuitivo, la consulta de tipo declaración en SQL-incluso un usuario familiarizado con las bases de datos se pueden utilizar para buscar datos.

CREACIÓN DE CONDICIONES.

Caín dijo antes, "ahora necesitamos aprender a como hacer las condiciones" vamos a ver algunas maneras de crear condiciones usando SQL.

OPERADORES DE COMPARACIÓN.

Una forma de expresar condiciones es mediante el uso de operadores de comparación como $>$ $=$ y $=$ por ejemplo, la condición de "A es mayor que o igual a B" se expresa usando $>=$, y la condición de "A es igual a B" se expresa usando $=$. Más ejemplos de operadores de comparación se muestran en la siguiente tabla.

OPERADORES DE COMPARACIÓN.

operadores de comparación.	Descripción	Ejemplo	Descripción del ejemplo
$A = B$	A es igual a B	precio_unidad=200	Precio por unidad es de 200g.
$A > B$	A es mayor que B	precio_unidad>200	Precio por unidad es mayor de 200g.
$A >= B$	A es igual o mayor que B	precio_unidad>=200	Precio por unidad es mayor o igual a 200g.
$A < B$	A es menor que B	precio_unidad<200	Precio por unidad es menor que 200g
$A <= B$	A es igual o menor que B	precio_unidad<=200	Precio por unidad es menor o igual a 200g.
$A <> B$	A es diferente de B	precio_unidad<>200	Precio por unidad diferente de 200g.

OPERADORES LOGICOS.

En algunos casos, es necesario expresar las condiciones de comparación que son más complejas que la simple. Se puede usar operadores lógicos (AND, OR y NOT) para combinar el operador basado en las condiciones y crear condiciones más complicadas, como se muestra en la siguiente tabla.


OPERADORES LÓGICOS.

Operadores lógicos	Descripción	Ejemplo	Descripción de ejemplo
AND	A and B	Codigó de prodcutu $>=200$ AND precio por unidad $=100$	El codigó del producto es mayor o igual a 200 y el precio por unidad es 100g.
OR	A or B	Codigó de producto $>=200$ OR precio por unidad $=100$	El codigó de producto es mayor o igual a 200 o el precio por unidad es 100g.
NOT	Not A	NOT precio por unidad $=100$	El precio por unidad no es 100g.

PATRONES.

Cuando no sabes exactamente lo que debes buscar, también puedes usar la comparación de patrones en las condiciones usando caracteres comodín. Cuando se usa la comparación de patrones use caracteres como `OR_%` en una declaración: Este buscará una cadena de caracteres que coincide con el patrón especificado. Puede buscar un valor que corresponde a una cadena de caracteres parcialmente especificado usando `%`, lo que indica una cadena de caracteres de cualquier longitud, `AND_`. Lo que especifica sólo un carácter.

Un ejemplo de una consulta mediante comodines se muestra a continuación. Este ejemplo afirma de búsqueda para una cadena de caracteres que tiene *n* en el extremo final del nombre de producto.

<pre>SELECT * FROM producto WHERE nombre_producto LIKE '%n';</pre>	 Esta es una sentencia para comparar patrones usando comodines.	<table><tr><th>Codigo producto</th><th>Nombre producto</th><th>Precio unidad</th></tr><tr><td>101</td><td>Melon</td><td>800G</td></tr><tr><td>104</td><td>Lemon</td><td>200G</td></tr></table>	Codigo producto	Nombre producto	Precio unidad	101	Melon	800G	104	Lemon	200G
Codigo producto	Nombre producto	Precio unidad									
101	Melon	800G									
104	Lemon	200G									

Los comodines se pueden utilizar en una sentencia de SQL que se explica a continuación.

COMODINES.

Comodín	Descripción	Ejemplo de patrón	Comparacion de cadena de caracter

%	Comparar algun número de caracteres	%n N%	Limón Melón Nuez hombligo de naranja
_	Compara un caracter	_t _t_	It To

BUSQUEDAS

Hay muchos métodos de búsqueda. Por ejemplo, puede especificar `BETWEEN X y Y` para el rango de un valor. Si especifica un rango como se muestra a continuación, se puede extraer productos con precio por unidad mayor o igual a 150G o menor que 200.

<pre>SELECT * FROM producto WHERE PRECIO_UNIDAD BETWEEN 150 AND 200;</pre>	 Especifica un rango de busqueda
--	---

Además, puede especificar si es `NULL` cuando la busqueda sea por filas. Si usa la búsqueda se muestra a continuación, se puede extraer productos con precios nulos.

<pre>SELECT * FROM producto WHERE precio_unidad es NULL;</pre>	 Búsqueda por un nulo.
--	--



PREGUNTAS

Ahora vamos a crear las sentencias SQL usando varios tipos de condiciones. vamos a usar la tabla destino de las exportaciones siguiente (suponiendo que la unidad de población es de 10.000). Conteste los siguientes preuntas usando sentencias SQL. Las respuestas están en la página 119.

Tabla de destino de exportación

Codigo destino de exportación	Nombre destino de exportación	Población
12	El Reino de Minanmi	100
23	Imperio Alfa	120
25	El Reino de Ritol	150
30	El Reino de Sazanna	80

Q1

Encontrar países en los que la población es mayor o igual a 1 millón, extraer de la tabla de abajo.

Codigo destino de exportación	Nombre destino de exportación	Población
12	El Reino de Minanmi	100
23	Imperio Alfa	120
25	El Reino de Ritol	150

Q2

Encontrar países en los que la población sea menor a 1 millón, extraer de la siguiente tabla

Codigo destino de exportación	Nombre destino de exportación	Población
30	El Reino de Sazanna	80

Q3

Encontrar países en los que el código destino de exportación es menor que 20 y la población es mayor que o igual a 1 millón.

Q4

Encontrar países en los que el código destino de exportación es mayor o igual a 30 y la población es mayor que 1 millón.

Q5

¿cuál es la población del reino de Ritol?

Q6

encontrar países cuyos nombres contienen la letra n.

FUNCIONES DE AGREGACIÓN



Princesa Ruruna y Cain han aprendido acerca de varias funciones de agregación. Las funciones de agregación también se conocen como *funciones de ajustes*. Puede utilizar estas funciones para agregar información como los valores máximo y mínimo. Número de artículos y suma.

Si especificas una frase WHERE junto con una función de agregación, se puede obtener un valor agregado sólo para las filas especificadas. Si se especifica una frase como la que se muestra a continuación, se puede averiguar el número de precios unitarios productos mayores o iguales a 200 g.

```
SELECT COUNT (*)  
FROM producto WHERE  
precio_unidad >= 200;
```



COUNT(*)
2

AGREGAR DATOS POR AGRUPACIÓN

Si datos de grupo, se puede obtener valores agregados fácilmente. Por ejemplo, si quieres obtener el número de productos y el promedio del precio unitario basado en la región, se puede utilizar las funciones de agrupación. Para agrupar datos combinar la función de agregación y la frase GROUP BY. vamos a utilizar la tabla de productos que se muestra a continuación.

TABLA PRODUCTO

Código producto	Nombre producto	Precio unidad	región
101	Melón	800G	Mar Sur
102	Fresa	150G	Centro
103	Manzana	120G	Mar Norte
104	Limón	200G	Mar Sur
201	Castaña	100G	Mar Norte
202	Melocotón	160G	Centro
301	Pera	130G	Mar Sur
302	Kiwi	200G	Mar Sur

para obtener el promedio del precio unitario de cada distrito en la Tabla producto, especificar la columna de región y la función AVG y la frase GROUP BY. Esto agrupará datos en función de la región y da el valor promedio unitario de los productos en cada región.

```
SELECT región, AVG (precio_unidad)  
FROM producto  
GROUP BY región;
```



Permite agrupar

región	AVG(precio_unidad)
Mar Sur	332.5
Mar Norte	110
Centro	155

¿Que si quiere restringir aún más los resultados, basado en una propiedad particular de los datos? supongamos que quiere encontrar productos regionales con el promedio de precios unitarios mayor o igual a 200g. En este caso, no especifique una condición con la frase WHERE, pero en su lugar use la frase HAVING. Esto le permite extraer sólo las regiones con el precio unitario promedio que es mayor o igual a 200 g.



PREGUNTAS.

Responder las siguientes preguntas usando la tabla destino de exportación (asumiendo que la unidad de población es 10.000). la respuesta está en la página 120.

Tabla destino de esportación

Codigó destino exportación	Nombre destino exportación	Población	Región
12	Reino de Minanmi	100	Mar Sur
15	Reino de Paronu	200	Centro
22	Reino de Tokanta	160	Mar Norte
23	Imperio Alfa	120	Mar Norte
25	Reino de Ritol	150	Mar Sur
30	Reino de Sazanna	80	Mar Sur
31	Reino de Taharu	240	Mar Norte
33	Reino de Maryon	300	Centro

Q7

¿cuál es la población más pequeña?

Q8

¿cuál es la población más grande?

Q9

¿cuál es el total de la población de todos los países incluidos en la tabla destino de exportación?

Q10

¿Cuál es la población total de los países en los que el código de destino de las exportaciones es mayor que 20?

Q11

¿Cuántos países hay que la población es mayor o igual a 1 millón?

Q12

¿cuántos países se encuentran en la región de Mar Norte?

Q13

¿qué país, en la región norte del mar tiene la población más grande?

Q14

¿Cuál es la población total de cada país, con excepción del Reino de Ritol?

Q15

Encontrar las regiones en los que el promedio de la población es mayor o igual a 2 millones.

Q16

Encontrar las regiones que contienen al menos tres países.

BUSCANDO DATOS SIN PARTÍCULA LA



Hay métodos de consulta más complejas disponibles en SQL, además de los que ya hemos discutido.

UTILIZANDO UNA SUBCONSULTA

Por ejemplo, puede incrustar una consulta en otra consulta. Esto se llama una *subconsulta*. echemos un vistazo a la tabla de abajo.

TABLA PRODUCTO

Codigó producto	Nombre producto	Precio unidad
101	Melón	800G
102	Fresa	150G
103	Manzana	120G
104	Limón	200G

TABLA DECLARACION VENTA

Codigó reporte	Codigó producto	Cantidad
1101	101	1.100
1101	102	300
1102	103	1.700
1103	104	500
1104	101	2.500
1105	103	2.000
1105	104	700

Puede utilizar estas dos tablas para buscar los nombres de productos de los que el volumen de ventas es mayor que o igual a 1000. la siguiente sentencia SQL llevará a cabo esa búsqueda.

```
SELECT * FROM producto
WHERE codigó_producto IN
(SELECT producto_codigó
FROM declaracion_venta
WHERE cantidad >= 100);
```

esta declaración contiene una subconsulta.

En esta sentencia SQL, la instrucción SELECT en paréntesis se lleva a cabo primero: el código de producto en la tabla de declaración de venta se busca primero, y los códigos de producto 101 y 103 se encuentran (ya que estos son los únicos informes con un volumen de ventas mayor a 1000). estos códigos de los productos se utilizan como parte de la condición para la instrucción SELECT fuera del paréntesis. Así, los productos que corresponden a los códigos de los productos 101 y 103 serán devueltos.

En otras palabras, en el caso de una subconsulta, el resultado de la instrucción SELECT entre paréntesis enviará la instrucción en otro SELECT para la búsqueda. La siguiente información será el resultado de la consulta total.

Código producto	Nombre producto	Precio unidad
101	Melón	800G
103	Manzana	120G

UTILIZANDO UNA SUBCONSULTA CORRELACIONADA

vamos a considerar una subconsulta como está contenida dentro de otra consulta. Una subconsulta puede hacer referencia a los datos de la consulta externa. Esto se llama una *subconsulta correlacionada*. En la siguiente consulta, la tabla declaracion_venta en la consulta externa es temporalmente debido a un nombre nuevo U para la subconsulta puede referirse a una ambigüedad. La sintaxis U.codigo_producto indica qué columna está destinada a codigo_producto, ya que hay dos fuentes de esa columna dentro de la subconsulta. Debido a que la subconsulta se refiere a los datos de la consulta exterior, la subconsulta no es independiente de la consulta externa como en los ejemplos anteriores. Esta dependencia se denomina una *correlación*.

❶ SELECT *
FROM declaracion_venta U

❷ WHERE cantidad >

❸ (SELECT AVG (cantidad)
FROM declaracion_venta
WHERE codigo_producto=
U.codigo_producto);

Report code	Product code	Quantity
1104	101	2.500
1105	103	2.000
1105	104	700

Esta consulta extrae declaraciones con Mayor volumen de ventas que el promedio de los productos.

Echemos un vistazo a cómo esta subconsulta correlacionada se procesa. En la subconsulta correlacionada, la consulta exterior se implementó primero.

❶ SELECT *
FROM declaracion_ventas U

Este resultado se envía al interior de la consulta para ser evaluada fila por fila. vamos a explorar la evaluación de la primera fila, código de producto 101.

❶

```
SELECT AVG (cantidad)
FROM declaracion_ventas
WHERE codigo_producto=101
```

El código de producto para la primera fila es de 101, o melones: la cantidad promedio de venta de melones es de 1800. Este resultado se envía como una condición para la consulta externa.

❷

```
WHERE cantidad >(1,800)
```

Este proceso continúa para todas las filas en la declaracion de ventas, los pasos 2 y 3 se llevan a cabo para todos los posibles códigos de producto. En otras palabras, esta consulta extrae informes en los que el volumen de ventas de una fruta es mayor que la cantidad promedio particular de fruta en ventas. En consecuencia, sólo las filas quinta, sexta y séptima del paso 1 son extraídas.



PREGUNTAS.

Ahora, conteste las siguientes preguntas basadas en la tabla de productos y la tabla declaración de ventas. Las respuestas en la página 122.

Q17

Encontrar la declaración de ventas por fruta con precios unitarios mayores o iguales a 300 g, y se extraer de la tabla de abajo.

Código reporte	Código producto	Cantidad
1101	101	1.100
1104	101	2.500

Q18

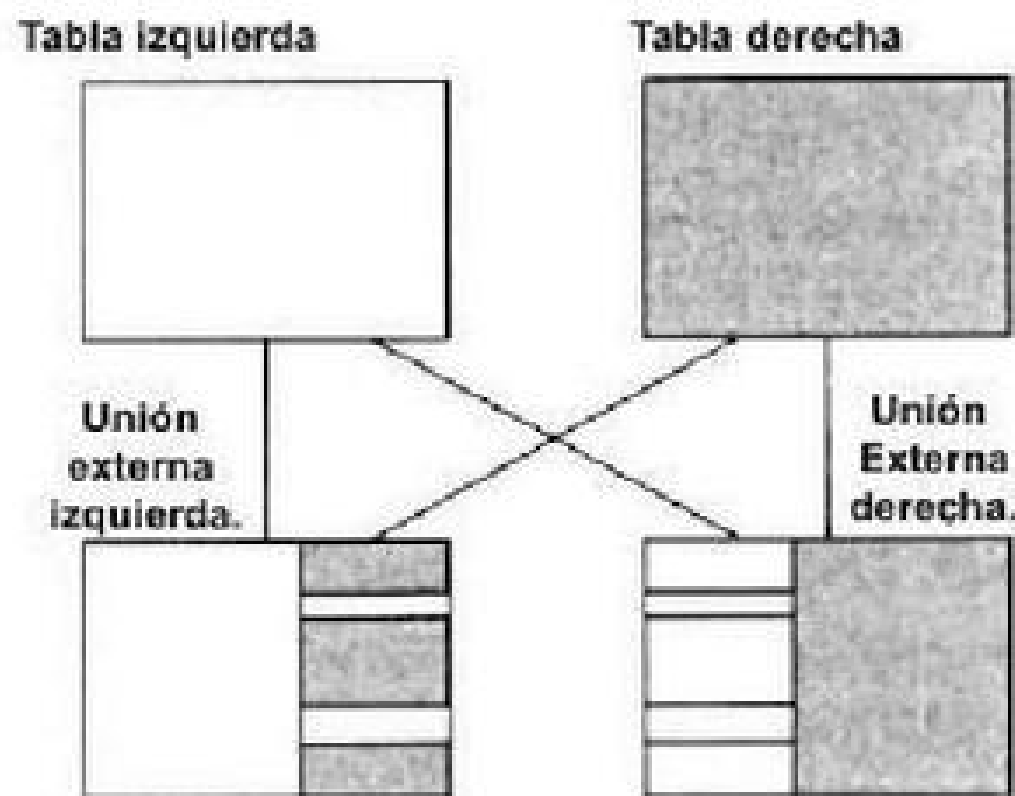
Obtener el volumen promedio de ventas por producto, y encontrar los artículos que tienen volúmenes de ventas que son menores al promedio.

UNIENDO TABLAS

Después de realizar una búsqueda basada en SQL, princesa Ruruna y Caín crearon un informe de ventas combinando tablas. Uniendo tablas por la combinación de las columnas con los mismos nombres es llamada una unión equi. Para una unión equi, las filas con el mismo valor son designadas como condiciones de unión para unir tablas. Uniendo columnas con el mismo nombre en uno son llamadas una unión Natural.

El método de unión en el que sólo las filas que tienen un valor en común unión igual son seleccionadas esto es llamado unión interna.

En contraste, el método de unión que mantiene todas las filas de una tabla y especifica un valor nulo para las filas no incluido en otra tabla es llamada unión externa. Si coloca una tabla creada a partir de una unión externa a la derecha o a la izquierda de una sentencia de SQL, es llamada una unión externa izquierda o una unión externa derecha, dependiendo de las filas que se mantienen.



CREANDO UNA TABLA



Finalmente, la princesa Ruruna y Caín han aprendido acerca de la sintaxis de la sentencia que crea una tabla. **CREATE TABLE**. La sintaxis de la sentencia dentro de una sentencia **CREATE TABLE** con frecuencia depende del tipo específico de base de datos que utiliza. Un ejemplo se muestra a continuación.

```
CREATE TABLE prodcuto
(
  codigo_producto int NOT NULL,
  nombre_producto varchar(255),
  precio_unidad int,
  PRIMARY KEY (codigo_producto)
);
```

Esta sentencia crea una tabla.

Cuando creas una tabla, debe especificar los nombres de las columna.

Adicionalmente, puede especificar una llave primaria y una llave secundaria para cada columna. En este ejemplo, el código del producto se especifica como una **LLAVE PRIMARIA** y código de producto no se le permite ser nulo. Cuando creas una tabla, puede que necesite incluir las siguientes especificaciones.

RESTRICCIONES EN UNA TABLA

Restricción	Descripción
LLAVE PRIMARIA	establece una llave principal
ÚNICO	debe ser único
NO NULO	no acepta un valor nulo
COMPROBAR	comprobar un rango
DEFECTO	establece un valor por defecto
REFERENCIAS DE LLAVE SECUNDARIA	establece una clave externa

Estas especificaciones son llamadas restricciones. Dando restricciones cuando se crea una tabla ayuda a prevenir conflictos de datos más adelante y le permite gestionar correctamente la base de datos.

INSERTANDO, ACTUALIZANDO O ELIMINANDO FILAS.

Puede utilizar la instrucción INSERT, UPDATE y DELETE para insertar, actualizar o eliminar datos de una tabla creada por la sentencia CREATE TABLE. Vamos a insertar, actualizar y borrar algunos datos utilizando SQL.

```
INSERT INTO producto
(codigo_producto, nombre_producto,
precio_unidad)
VALUES (200, 'cereza', 200);
```

← Esta sentencia añade cereza

```
UPDATE producto
SET nombre_producto= 'cantaloupe'
WHERE nombre_producto= 'melón';
```

← Esta sentencia actualiza Melón a cantaloupe.

```
DELETE FROM producto
WHERE nombre_producto='manzana';
```

← Esta sentencia elimina manzana

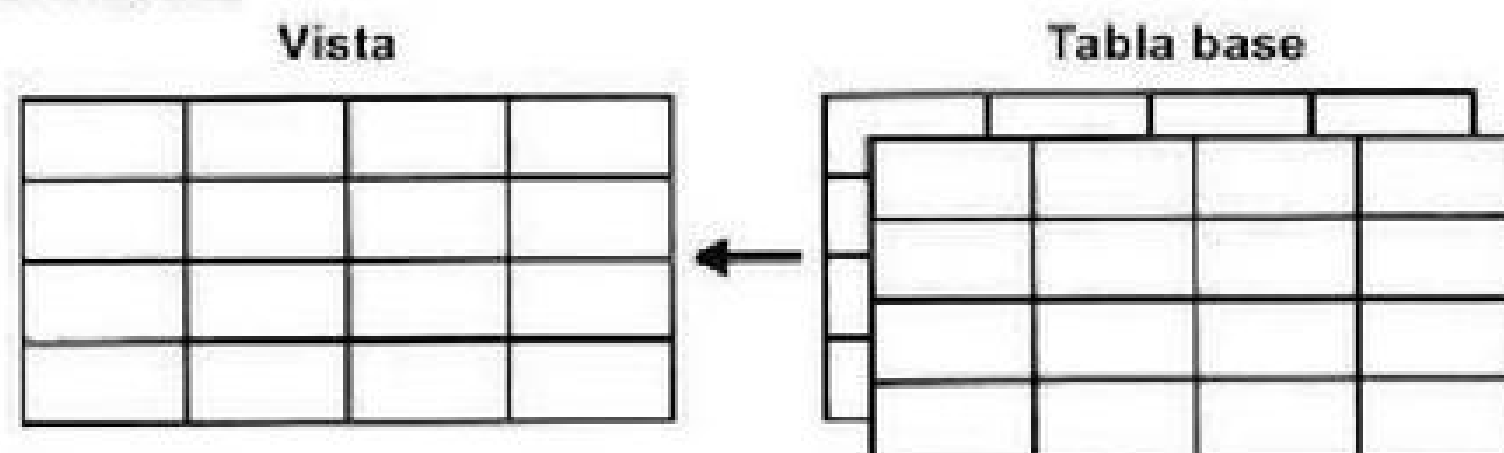
Codigo producto	Nombre producto	Precio unidad	
101	Cantaloupe	800G	Actualizó a cantaloupe.
102	Fresa	150G	
103	manzana	120G	Manzana fue eliminada
104	Limón	200G	
200	Cereza	200G	Cereza fue añadido

Insertando, actualizando o eliminando una fila, no puede violar las restricciones colocadas por la sentencia CREATE TABLE. Si un producto con el código de producto 200 ya existe, no se puede agregar la cereza, puesto que no puedes añadir datos duplicados como llave primaria. Cuando inserta, actualiza o elimina datos en una base de datos, debe tener en cuenta las restricciones de la base de datos.

CREANDO UNA VISTA



basado en la tabla que ha creado con la instrucción **CREATE TABLE**, también puede crear una tabla virtual que existe solo cuando es vista por un usuario. Esto se llama una *vista*. La tabla de la que se deriva una vista es llamada *tabla base*.



Utiliza la sentencia SQL que se muestra a continuación para crear una vista.

```
CREATE VIEW producto_caro
(codigo_producto, nombre_producto,
precio_unidad)
AS SELECT *
FROM producto
WHERE precio_unidad >= 200;
```

Esta sentencia crea una vista

La tabla de productos caros es una vista basada en la tabla de productos que es una tabla base. Fue creada por la extracción de datos con precio unitario mayor que o igual a 200g de la tabla producto.

TABLA DE PRODUCTO CARO

Código producto	Nombre producto	Precio unidad
101	melón	800G
104	Limón	200G
202	Caqui	200G

una vez que se crea la vista producto caro, puede buscar datos de la misma manera que debería para la búsqueda de datos en una tabla base.

```
SELECT *
FROM producto_caro
WHERE precio_unidad >= 500;
```

Permite que la vista sea usada de la misma manera como una tabla base.

Es conveniente al crear una vista cuando se quiere hacer parte de los datos en una tabla de base pública.

También hay instrucciones SQL para borrar una tabla base o vista. La instrucción usada para eliminar una tabla base o vista se muestra a continuación.

```
DROP VIEW producto_caro;
```

```
DROP TABLE producto;
```



PREGUNTAS

Crear instrucciones SQL para las siguientes preguntas (suponiendo que la unidad de población es 10,000). las respuestas están en la página 123.

Q19

La tabla de destino exportación siguiente fue creada usando una sentencia CREATE TABLE. Añadir los datos a continuación.

TABLA DESTINO EXPORTACION

Código destino exportación	Nombre destino exportación	Población	Región
12	Reino de Minanmi	100	Mar Sur
15	Reino de Paronu	200	Centro
22	Reino de Tokanta	160	Mar Norte
23	Imperio Alfa	120	Mar Norte

Q20

De la tabla de destino de exportación en Q19, cree una vista titulada país mar norte que muestra los países pertenecientes a la zona mar norte.

TABLA DESTINO EXPORTACION

Código destino exportación	Nombre destino exportación	Población
22	Reino de Tokanta	160
23	Imperio Alfa	120

Q21

Cambiar la población del Reino de Tokanta en la tabla destino de exportación a 1,5 millones.

Q22

En la tabla destino de exportación, eliminar todos los datos para el Reino de Paronu.

RESUMEN



- Puede utilizar las funciones de SQL para definir, operar y controlar datos.
- Para buscar datos utiliza una instrucción **SELECT**.
- Para especificar una condición, utilice una frase **WHERE**.
- Para insertar, actualizar y eliminar datos, use **INSERT**, **UPDATE** y **DELETE**.
- Para crear una tabla, utilice una instrucción **CREATE TABLE**.

RESPUESTAS

Q1

```
SELECT *  
FROM destino_exportacion  
WHERE población>=100;
```

Q2

```
SELECT *  
FROM destino_exportacion  
WHERE población<100;
```

Q3

```
SELECT *  
FROM destino_exportacion  
WHERE codigo_destino_poblacion<20  
AND población>=100;
```

Código destino exportación	Nombre destino exportación	Población
12	Reino de Minanmi	100

Q4

```
SELECT *  
FROM destino_exportacion  
WHERE codigo_destino_poblacion<30  
AND población>100;
```

Código destino exportación	Nombre destino exportación	Población
23	Imperio Alfa	120
25	Reino de Ritol	150
30	Reino de Sazanna	80

Q5

```
SELECT poblacion
FROM destino_exportacion
WHERE nombre_destino_exportacion='el reino de Ritol';
```

poblacion
150

Q6

```
SELECT *
FROM destino_exportacion
WHERE nombre_destino_exportacion LIKE '%n%';
```

Código destino exportación	Nombre destino exportación	Población
12	Reino de Minanmi	100
30	Reino de Sazanna	80

Q7

```
SELECT MIN(poblacion)
FROM destino_exportacion;
```

MIN(poblacion)
80

Q8

```
SELECT MAX(poblacion)
FROM destino_exportacion;
```

MAX(poblacion)
300

Q9

```
SELECT SUM(poblacion)
FROM destino_exportacion;
```

SUM(poblacion)
1.350

Q10

```
SELECT SUM(poblacion)
FROM destino_exportacion
WHERE codigo_destino_exportacion>20;
```

SUM(poblacion)
1,050

Q11

```
SELECT COUNT(*)
FROM destino_exportacion
WHERE poblacion>=100;
```

COUNT(*)
7

Q12

```
SELECT COUNT(*)
FROM destino_exportacion
WHERE region>='mar norte';
```

COUNT(*)
3

Q13

```
SELECT MAX(poblacion)
FROM destino_exportacion
WHERE region='mar norte';
```

MAX(poblacion)
240

Q14

```
SELECT SUM(poblacion)
FROM destino_exportacion
WHERE NOT (nombre_destino_exportacion='reino de Ritol');
```

SUM(poblacion)
1,200

Q15

```
SELECT SUM(poblacion)
FROM destino_exportacion
GROUP BY region
HAVING AVG(poblacion)>=200;
```

Región	AVG(poblacion)
Centro	250

Q16

```
SELECT region COUNT(*)
FROM destino_exportacion
GROUP BY region
HAVING COUNT(*)>=3;
```

Región	COUNT(*)
Mar Norte	3
Mar Sur	3

Q17

```
SELECT *
FROM declaracion_ventas
WHERE codigo_producto IN
(SELECT codigo_producto
FROM producto
WHERE precio_unidad>=300);
```

Q18

```
SELECT *
FROM declaracion_ventas U
WHERE cantidad<
(SELECT AVG(cantidad)
FROM declaracion_ventas
WHERE codigo_producto=U.codigo_producto);
```

Report code	Product code	Quantity
1101	101	1,100
1102	103	1,700
1103	104	500

Q19

```
INSERT INTO destino_exportacion(codigo_destino_exportacion,
nombre_destino_exportacion, poblacion, region)
VALUES(12, 'reino de Minanmi', 100, 'Mar Sur');
INSERT INTO destino_exportacion(codigo_destino_exportacion,
nombre_destino_exportacion, poblacion, region)
VALUES(15, 'reino de Paronu', 200, 'Centro');
INSERT INTO destino_exportacion(codigo_destino_exportacion,
nombre_destino_exportacion, poblacion, region)
VALUES(22, 'reino de Tokanta', 160, 'Mar Norte');
INSERT INTO destino_exportacion(codigo_destino_exportacion,
nombre_destino_exportacion, poblacion, region)
VALUES(23, 'Imperio Alfa', 120, 'Mar Norte');
```

Q20

```
CREATE VIEW pais_Mar_Norte(codigo_destino_exporacion,
nombre_destino_exportacion, poblacion)
AS SELECT codigo_destino_exportacion, nombre_destino_exportacion
poblacion
FROM nombre_destino_exportacion
WHERE region='Mar Norte';
```

Q21

```
UPDATE destino_exportacion
SET poblacion_150
WHERE nombre_destino_exportacion='Reino de Paronu';
```

Q22

```
DELETE FROM destino_exportacio
WHERE nombre_destino_exportacion='Reino de Paronu';
```

ESTANDARIZACIÓN DE SQL

SQL está estandarizado por la Organización Internacional de Normalización (ISO). En Japón, es estandarizado por la norma JIS (Estándares Industriales Japones). Otras normas incluyen SQL SQL92, establecido en 1992, y SQL99, establecido en 1999. Los productos de bases de datos relacionales están diseñados para que las consultas se pueden hacer de acuerdo con estas normas.

Algunos productos de bases de datos relacionales tienen sus propias especificaciones. Consulte el manual de instrucciones para el producto de base de datos para obtener información adicional.

5

¡VAMOS A UTILIZAR UNA BASES DE DATOS!



¿QUE ES UNA TRANSACCIÓN?



Actualmente,
debo darle las
gracias ..



Pero todavía
tenemos mucho
que aprender.

Por ejemplo, me maravilla
porqué una base de datos
todavía puede funcionar
cuando muchos usuarios
están accediendo a ella al
mismo tiempo.

Por lo demás,
el tema de la
seguridad
también es un poco
mi asunto.



Aparentemente,
tienes algunas
preocupaciones
sobre tu base de datos.



Algo.

Bien, para
entender mejor
los asuntos,

Yo tengo
hecha una
pequena
investigación.

Oh, ¿sí?



El título
de mi
presentación es:

¿cómo puede una
base de datos permitir
a un gran número de
usuarios acceder
simultáneamente?



OH!

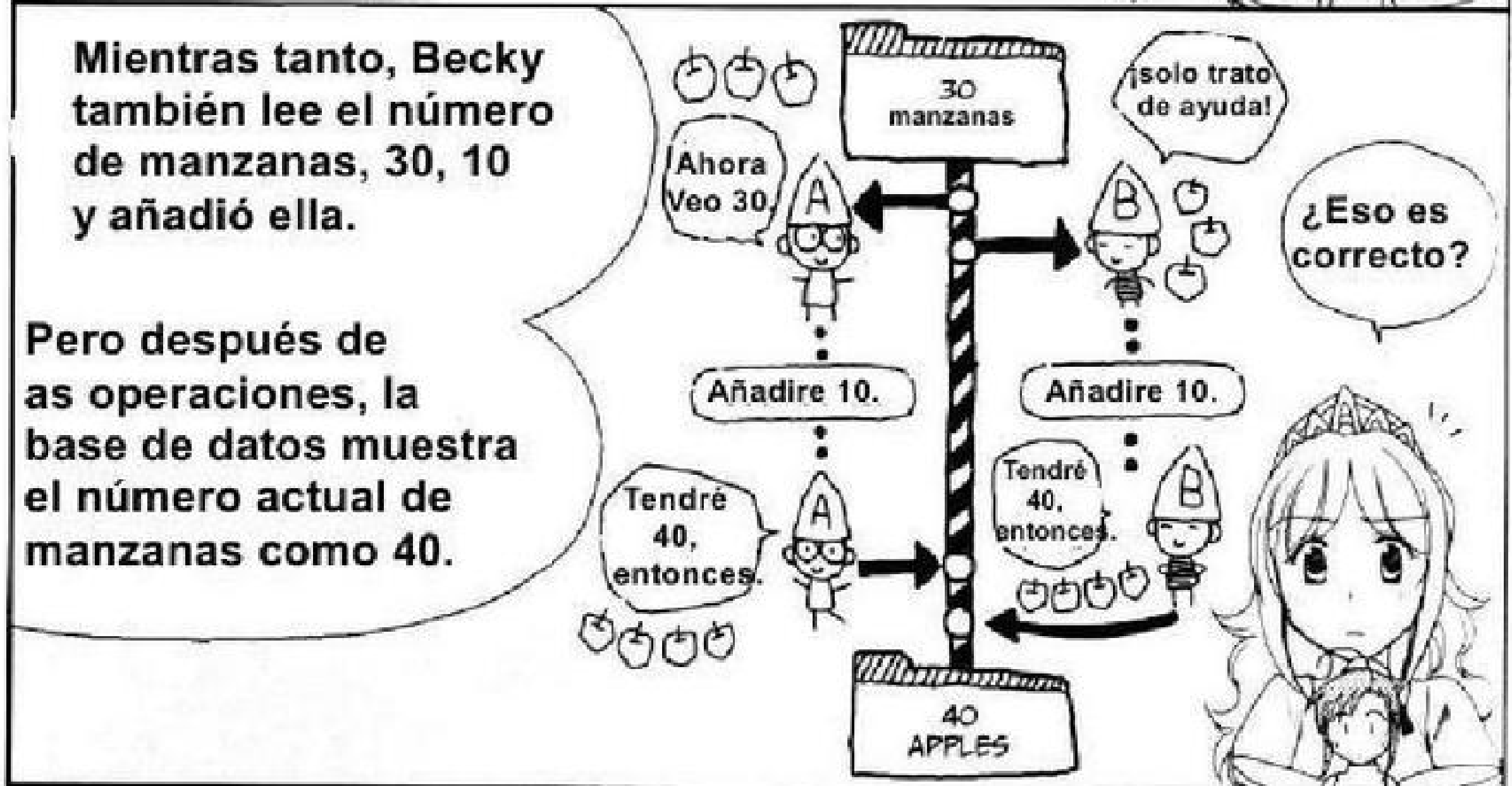
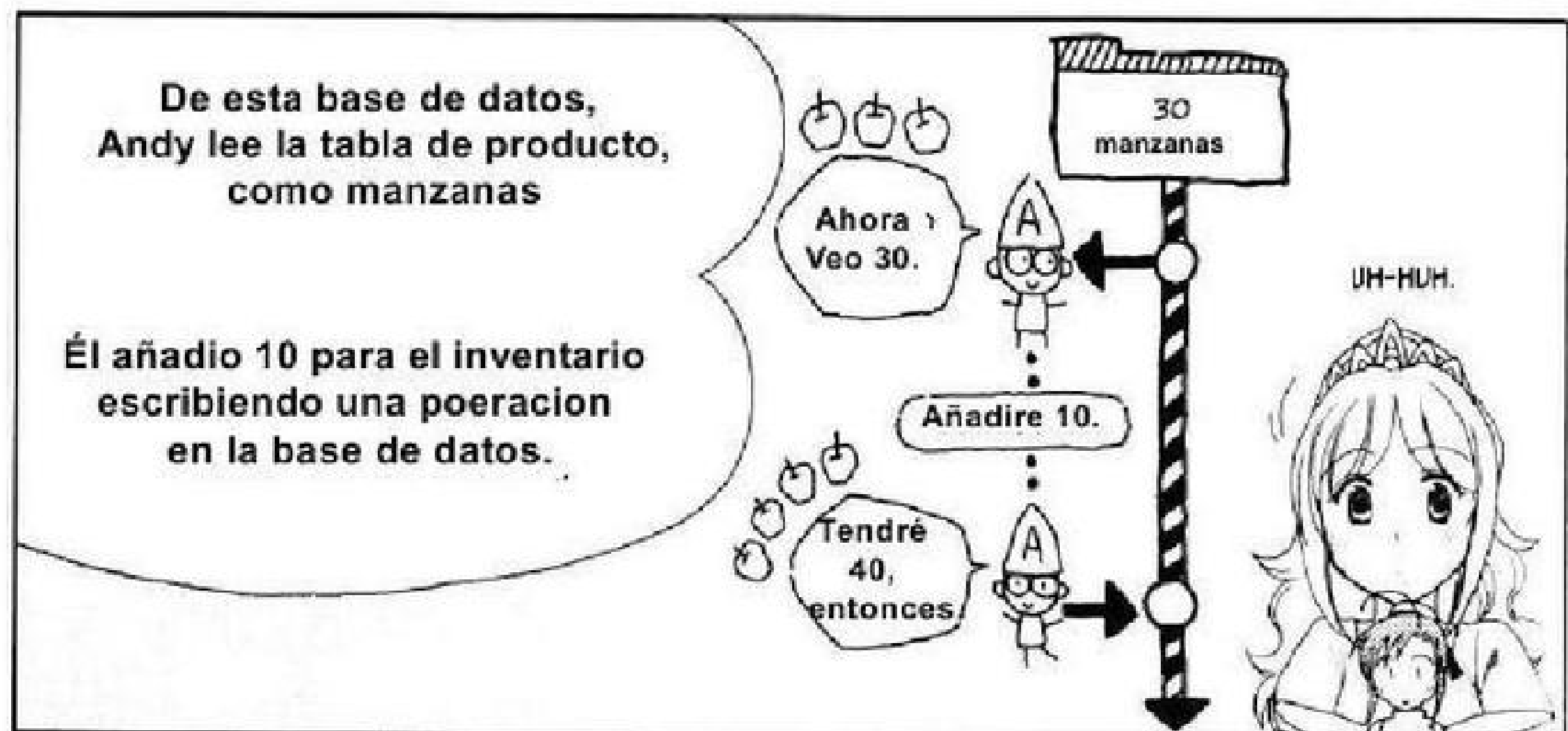
FLASH!

¡Yo tengo
Preprado un par
de ilustraciones
para ayudar a su
comprensión!

Vaya,
eso es genial.



¡Me encanta un
buen espectáculo!





¿QUE ES UN BLOQUEO?

En una base de datos,
las operaciones de
muchos usuarios son
controladas para
que nada salga mal...

Cuando ellos
accesan
a la base de
datos al
mismo tiempo.

Veo.

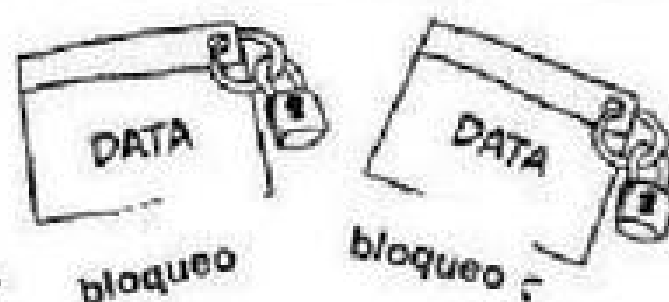
Con el proposito,
un método que es
Usado que es llamado
bloqueo.

Bloqueo

¿Te refieres a "bloqueo"
como en "bloqueo
y llave"?

Efectivamente.

Bloqueas dato
para prevenir que
comiensen a
procesarse
erroneamente.



Eso es nuevo
para mi.

Permiteme explicar
usando un ejemplo
anterior.

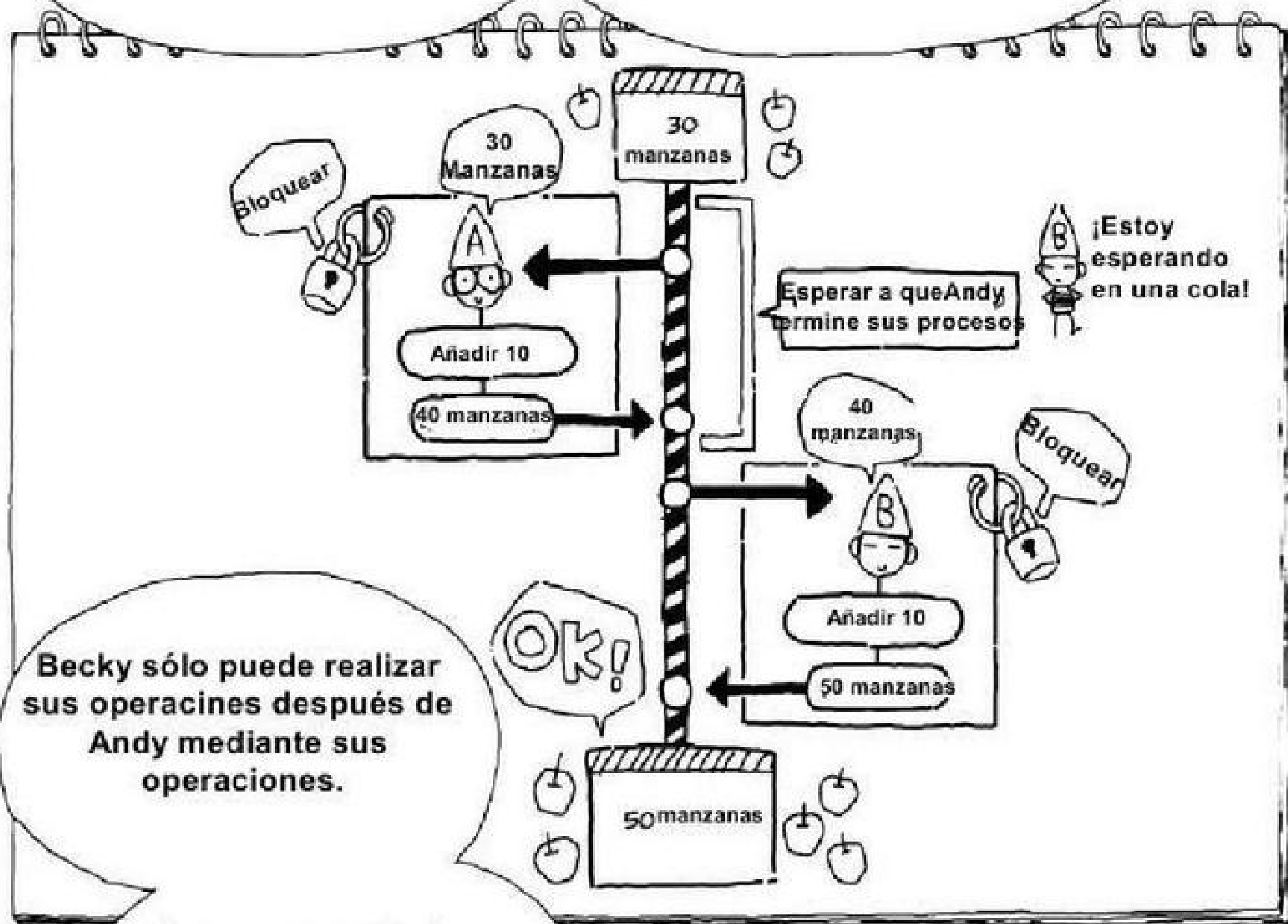
¡Tienes
Dibujos
de muchos
diagramas!

FLIP

FLIP

Andy bloquea los datos antes de realizar una serie de operaciones.

Cuando Becky trate de realizar sus operaciones, ella debe esperar hasta que Andy haya terminado.



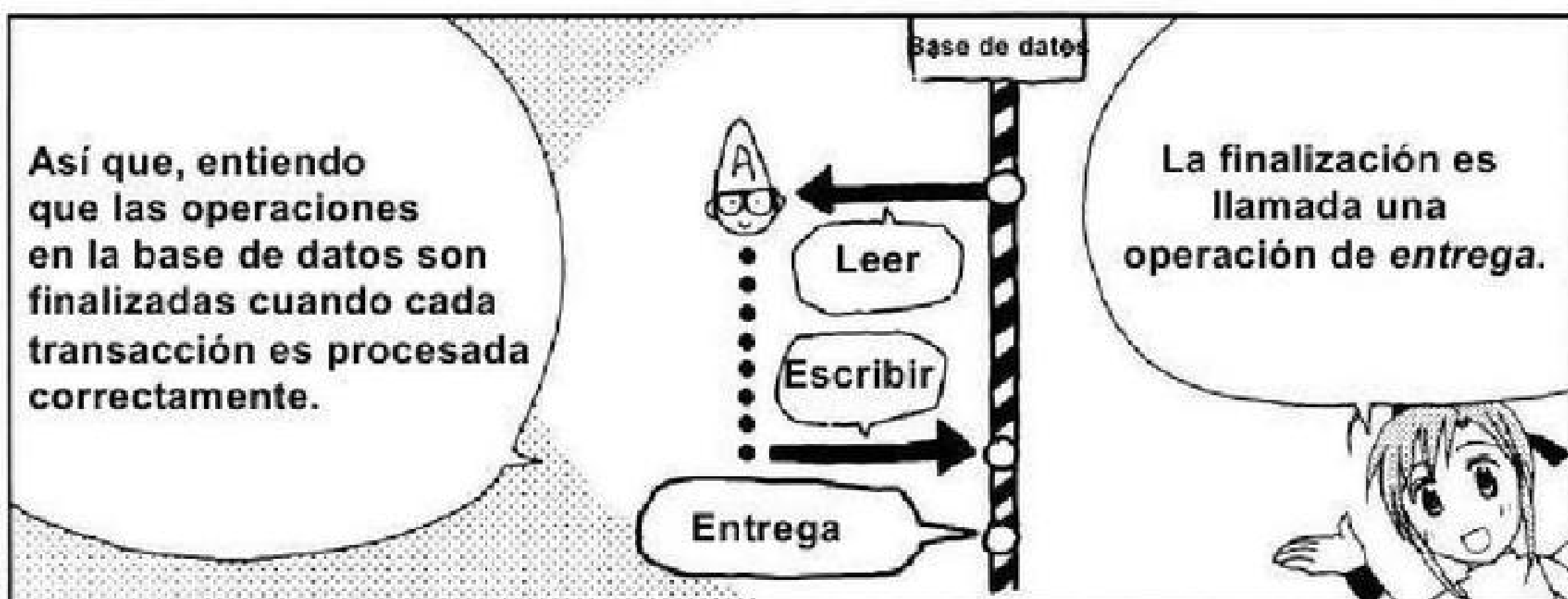
Becky sólo puede realizar sus operaciones después de Andy mediante sus operaciones.

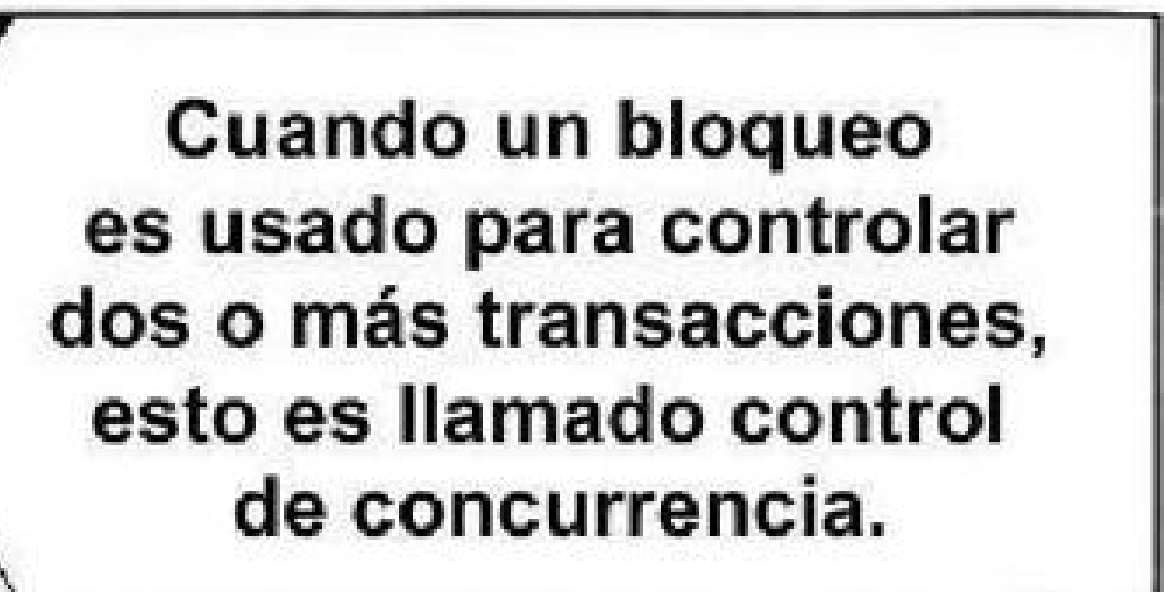
Como un resultado, la base de datos produce el valor de 50, como debe ser.

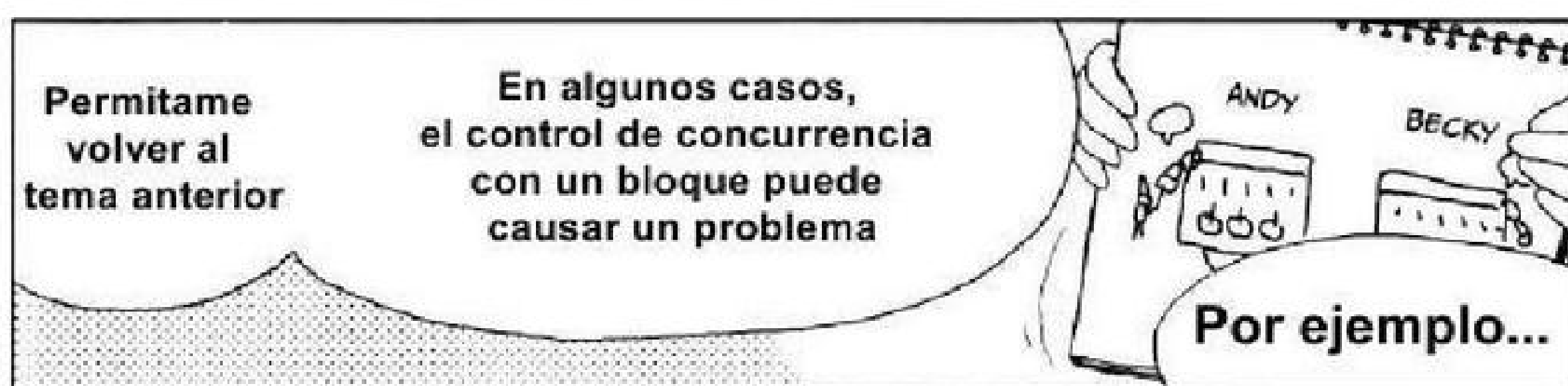
¡Oh, muchacho! Caín estoy impresionada

Sí, sí.













DATABASE SECURITY



Nuestra tabla de
Producto.



Código producto	Nombre producto	Precio unidad
101	Melón	10,000G
102	fresa	12,500G
103	Manzana	8,000G
104	Limón	6,000G
201	Castaño	9,000G
202	Caqui	12,400G
301	Pera	5,000G
302	Kiwi	6,000G

¿Qué hay de
malo con eso?



Los precios.
¡Los precios!



¿Los precios?



OH NO!!!



¡Las cifras en la
columna de precio
unidad todas estan
confundidos!

¡¡Qué caro
un melón!!

¿¿Cómo
Sucedio?!





Porqué sus facturas son un desastre, mi país, un importador, está experimentado en desastres.

Una base de datos es una cosa peligrosa.

Como parte de la compensación por este problema...



¿Porqué no aceptas mi propuesta, princes Ruruna? Ven a mi país y se mi novia.



ignorando



Alguien con malicia podría haber intentando realizar una sobre escritura de datos no autorizada.

¡Qué terrible!

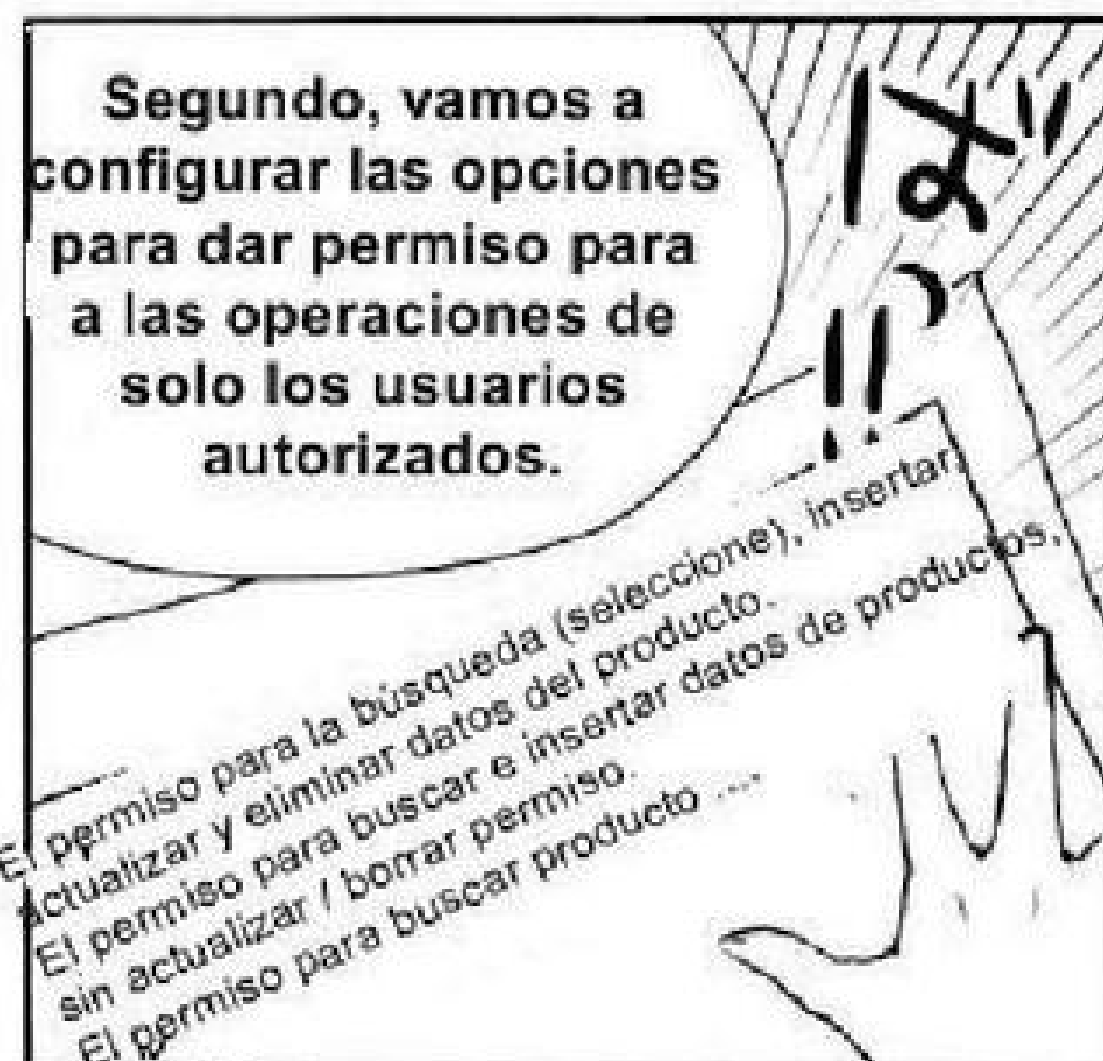
Te estás comportando como si yo no estuviera aquí.



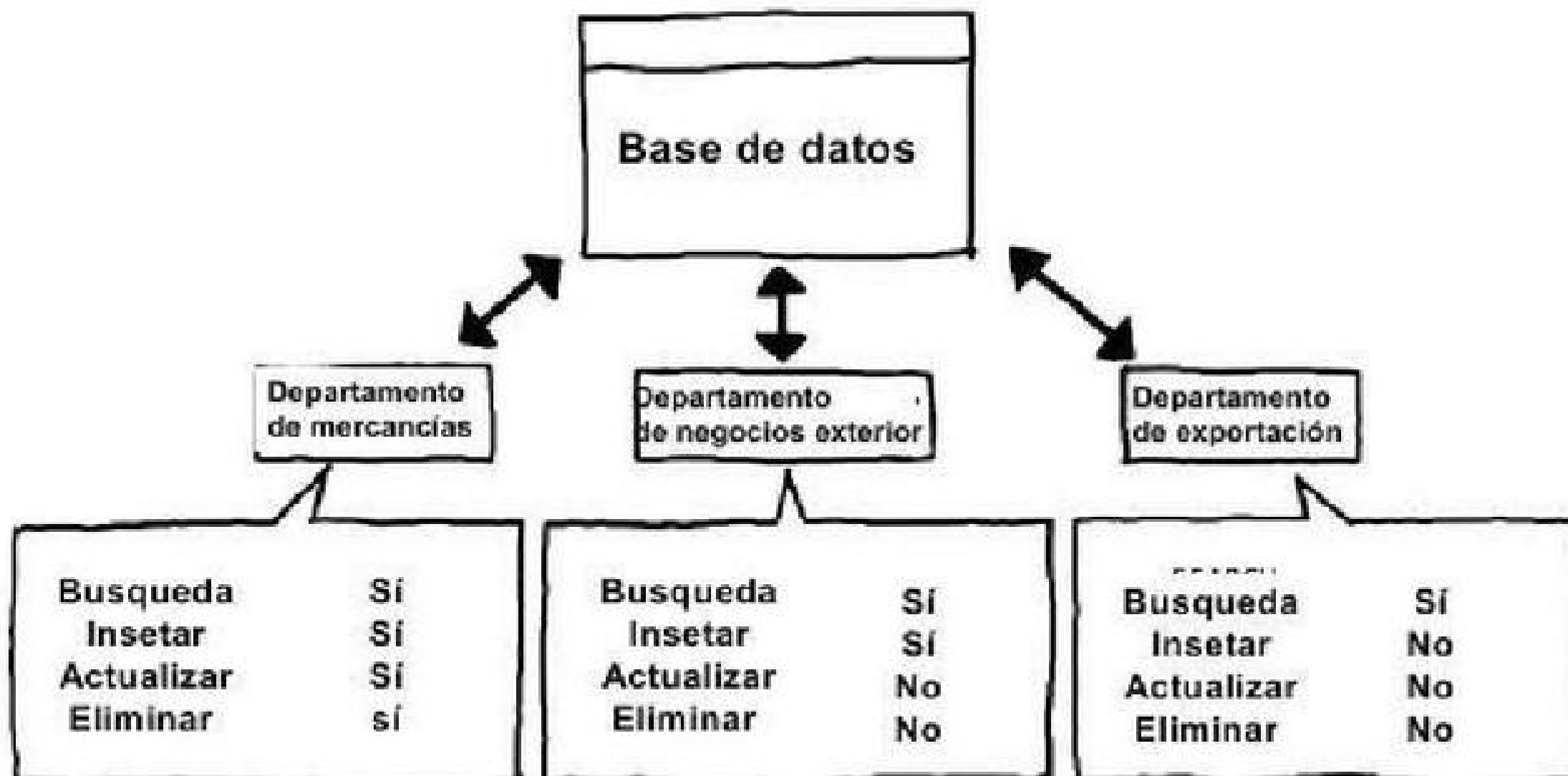
Príncipe Raminess,



Lo sentimos mucho.



- Personal de departamento mercancías puede buscar, insertar, actualizar y eliminar datos del producto.
- Personal de departamento de negocios en el extranjero puede buscar e insertar un dato de productos, pero no están autorizados para actualizar o eliminarlos.
- Personal de departamento Exportación puede buscar los datos del producto, pero no se les permite insertar, actualizar o eliminar.



हां ————— ्ह!!

No solo restringiremos el número de usuarios además los permisos que permitiremos para cada usuario con acceso a la base de datos.

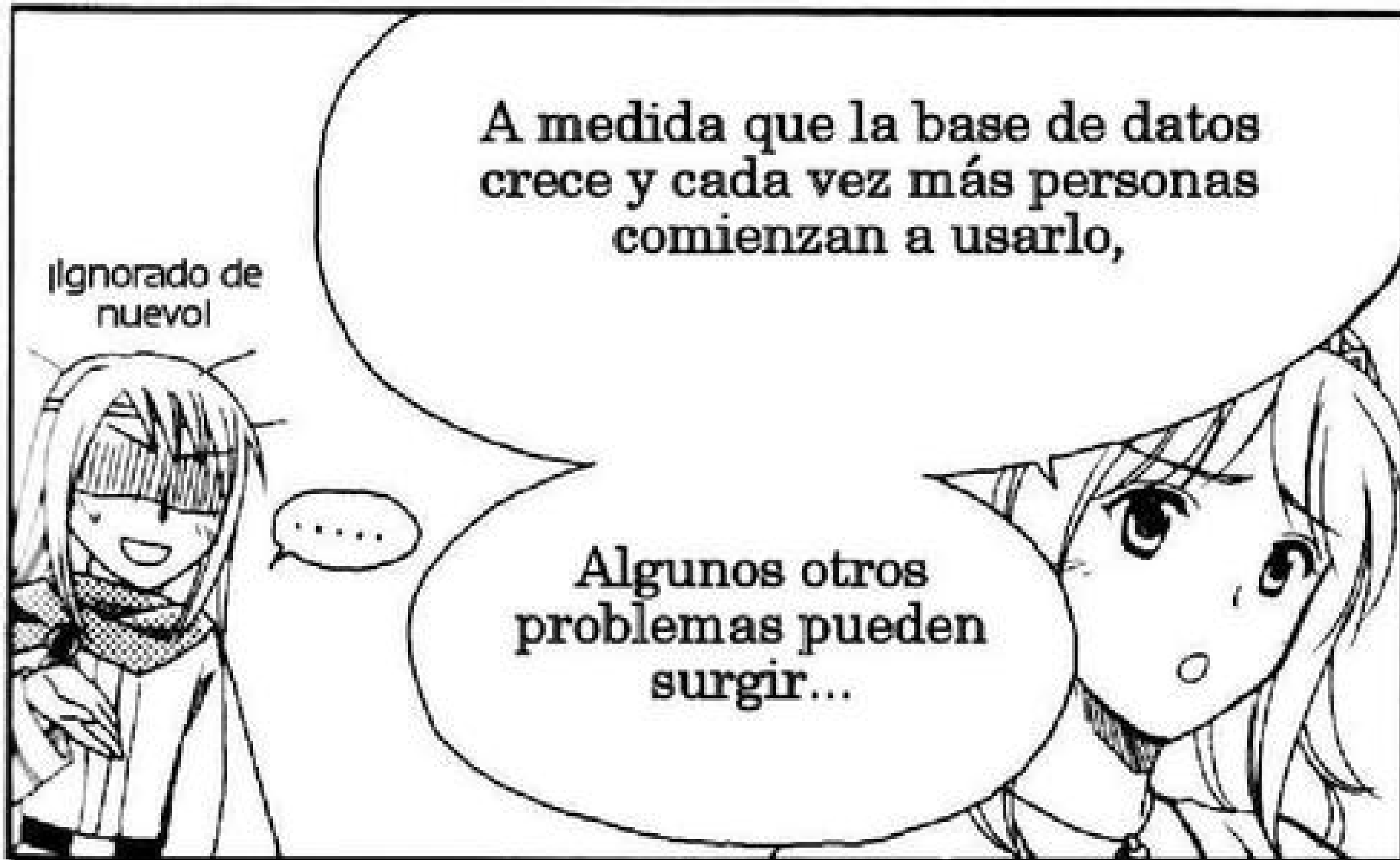
De esta manera, los problemas pueden ser evitados y la base de datos aún puede ser compartida.

Oh, sí...
Dejando eso de lado lo que quería decir es...

Eso es una buena oportunidad para que consideres casarte con migo, y

¡Oh, pero espera..!

Acelerando las cosas con la indexación



Supongamos que vamos a buscar el significado del término transacción mediante la consulta en el libro de bases de datos.

Una búsqueda ciega por el término sería laborioso, por lo que echaría un vistazo al índice, en su lugar.

Una búsqueda a ciegas por el término sería laborioso, por lo que echaría un vistazo al índice, en su lugar.

Indice

Página de referencia

Mediante el uso de un índice, que ¡puede encontrar rápidamente la página que está buscando!

Tienes razón.

Es lo mismo para un índice de base de datos. Por ejemplo...

Si se crea índices
para el código de
producto,

En cualquier momento puede
saber dónde se almacena los
datos del producto para un
producto asignado con código
de producto 101.



Le indica en qué parte del
disco se encuentra los
datos del producto.



Indexación ayuda a
acelerar la búsqueda.

Bueno, no es tan
fácil para mí seguir...
En fin...

UH-HUH.



Se requiere mucho tiempo
para buscar todas las filas
en la búsqueda de ciertos
datos.

¿Tengo que
repetir la
explicación
desde el
principio?

NO,
NO...



Usando los índices,
podemos reducir el
número de acceso al
disco.

¿Qué es
eso?

Reducir el número de
acceso a disco, y
nuestra búsqueda
¡será mucho más
rápido!

UH-OH.



¡Hey! ¿A quién
le hablas?

¿Hay alguien
aquí?

Princesa,
¡ten
cuidado!

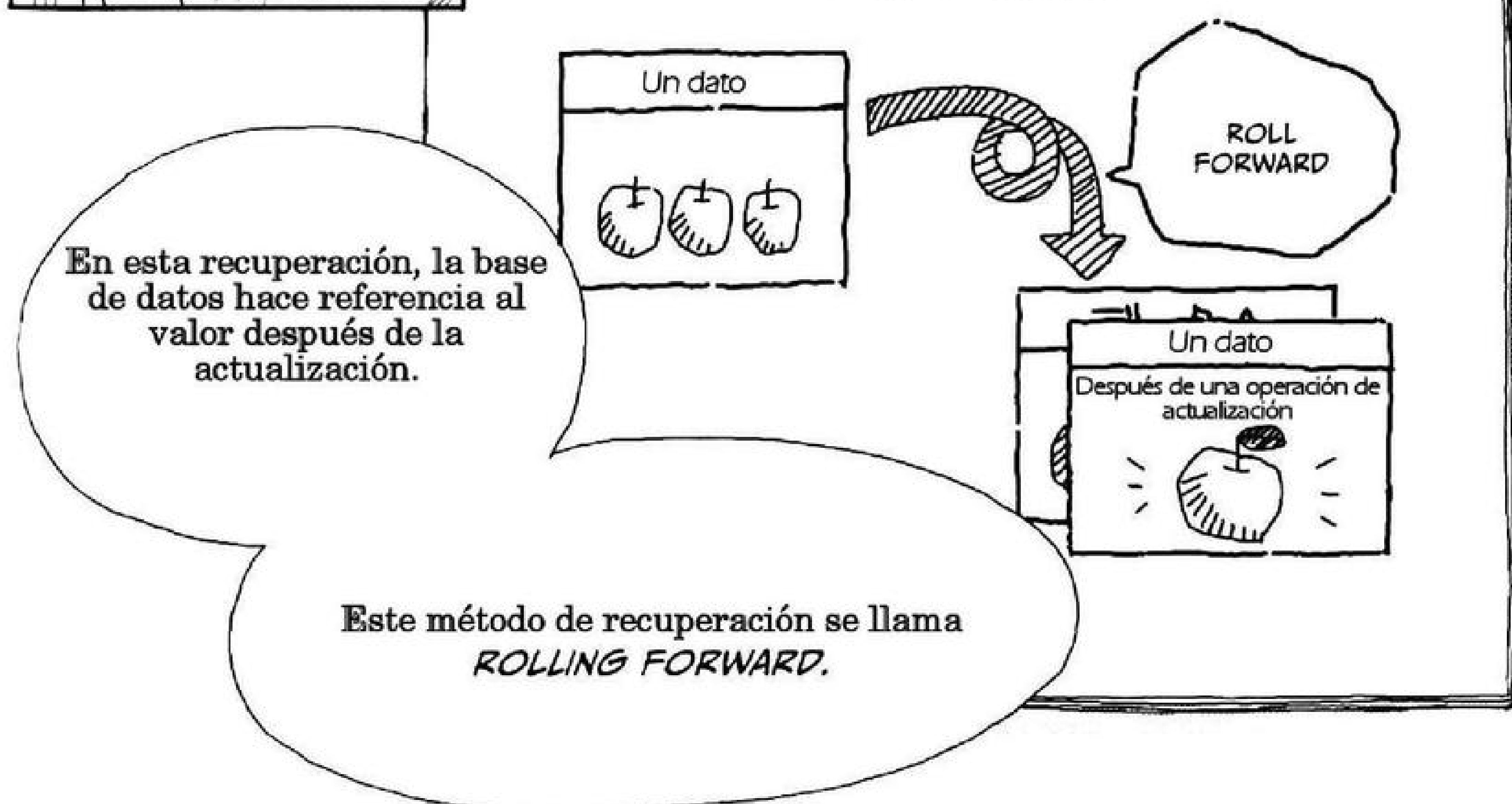
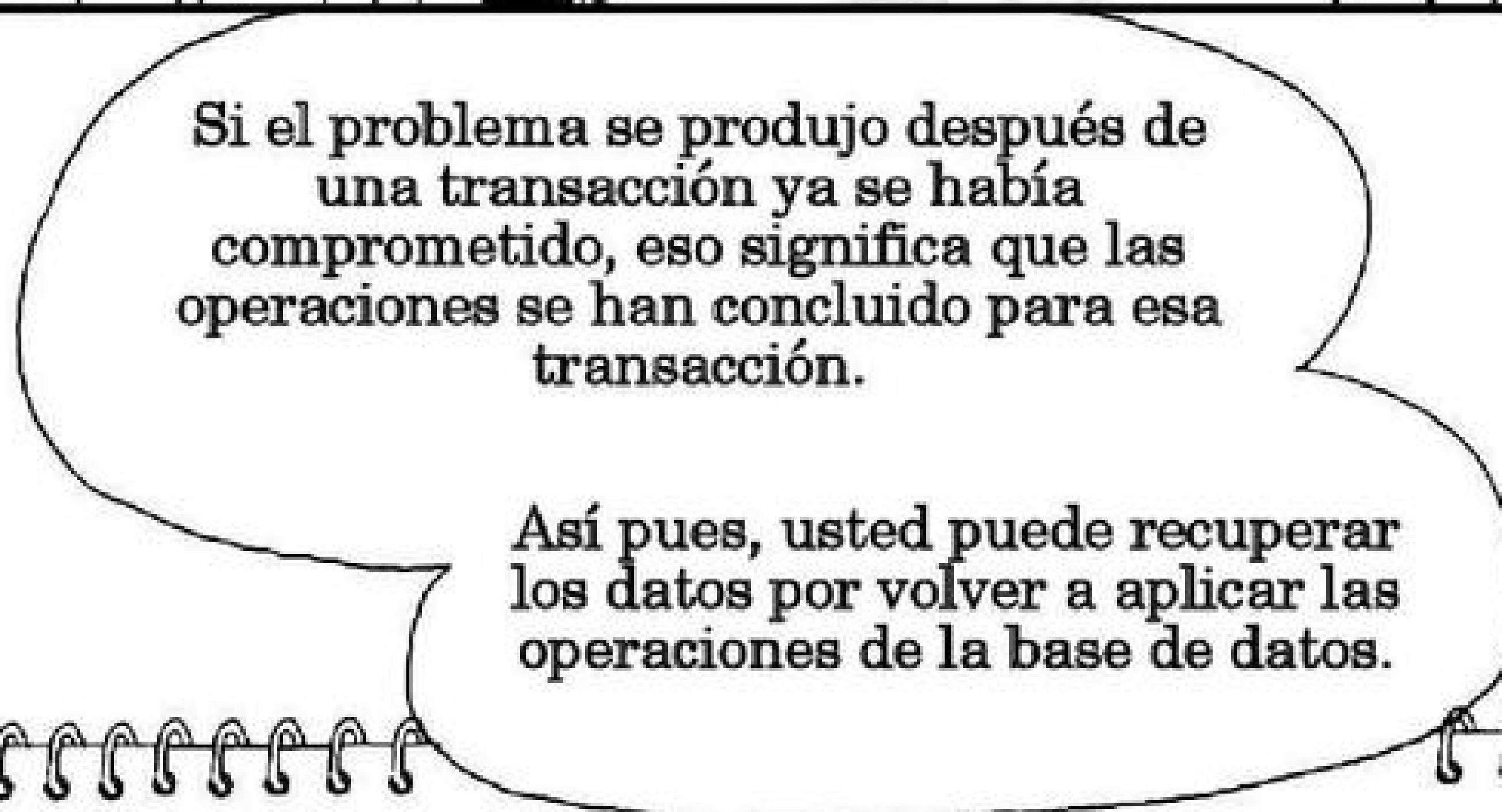


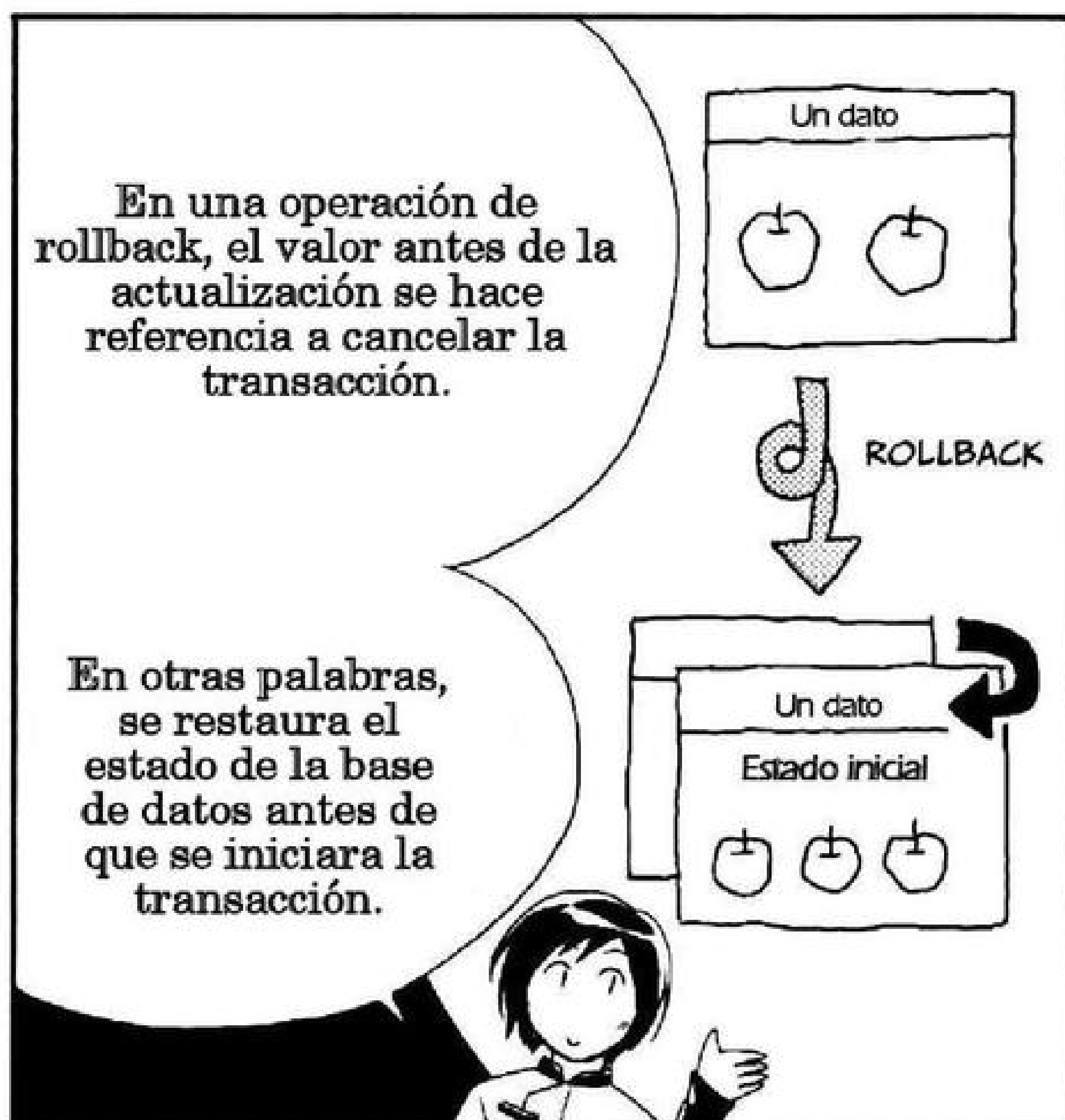




RECUPERACIÓN DE DESASTRES







Pues bien, teniendo en cuenta todo eso, le permitiré salirse con la suya esta vez.

Por Dios...

Pero...

Lo digo en serio acerca de nuestro matrimonio, ¿de acuerdo?

GULP

RURUNA...?

Raminess, lo siento.

No puedo aceptar tu propuesta.

¿Por qué no?

Debido a que...

¡Amo a alguien más!

Me pregunto quién...

¿No lo sabes...?

Mirando apasionadamente

OH!

SQUEEZE!

¡Aprieta!



Estaré con Caín para siempre, y con el poder de nuestra base de datos,

¡Vamos a ver que El Reino de Kod prosperará!

¿Qué demonios?
No... no me
hagas esto...

Lo siento.



¿Prefiere a
alguien tan
humilde como
Caín?

¡Bueno, sí,
quiero decir,
perdoname!



¿Por qué te
disculpas
Caín?

Oh, no, yo
no lo hago...
yo lo siento.



Quédate
conmigo
para
siempre,
Caín.



¡Sí, sí, su
alteza!



¿Por qué, oh, por
qué?



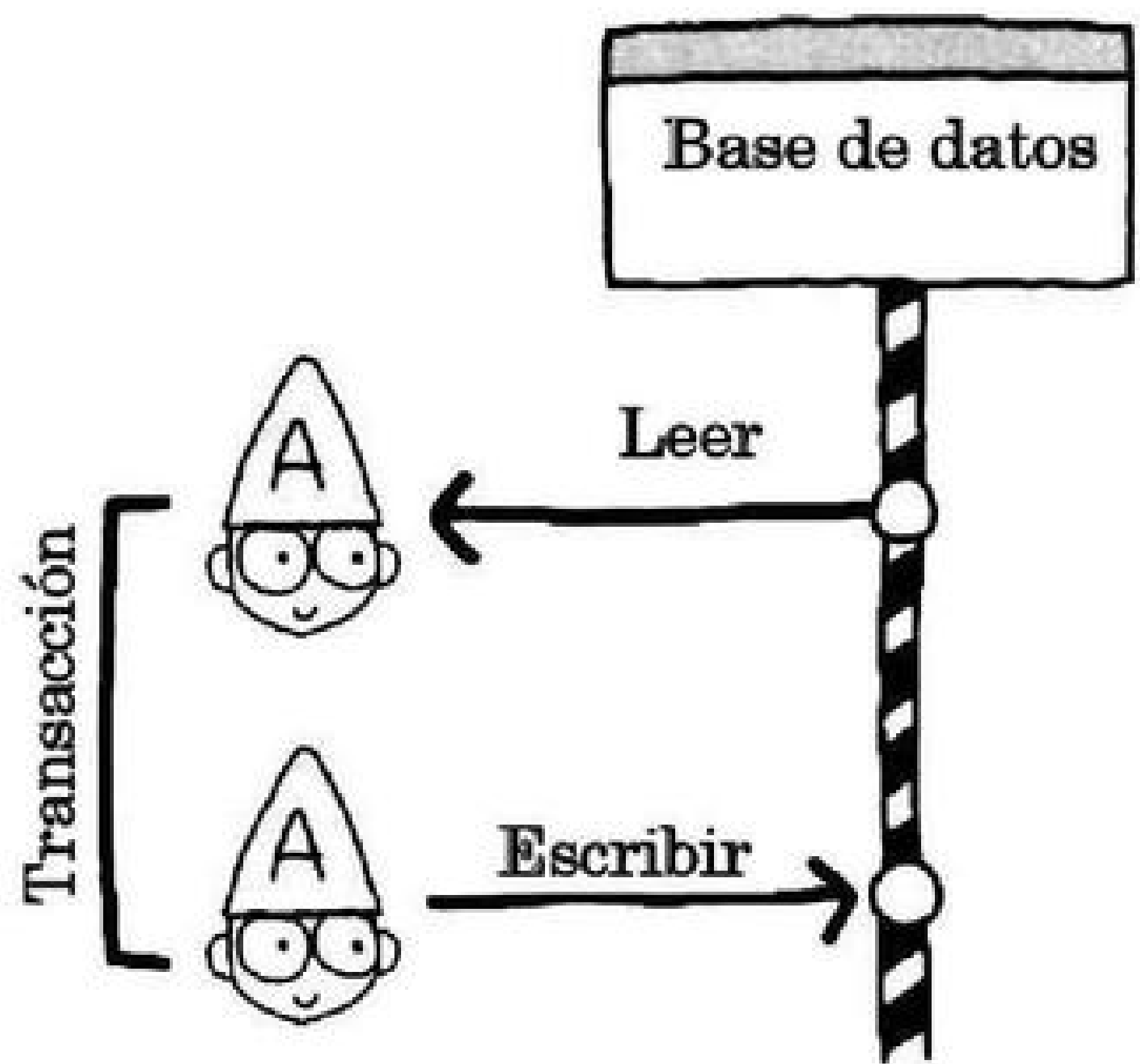
Ellos hacen
una gran
pareja.



PROPIEDADES DE LAS OPERACIONES



La investigación de Caín demostró que los usuarios de una base de datos puede buscar, insertar, actualizar y eliminar datos. Un conjunto de éxito de las operaciones realizadas por un solo usuario se llama una transacción.



Cuando el usuario comparte una base de datos, es importante asegurar que las transacciones múltiples pueden ser procesadas sin causar datos contradictorios. También es importante para proteger los datos de las inconsistencias en el caso de un fallo se produce mientras una transacción está siendo procesada. Con ese fin, la siguiente tabla se muestra las propiedades requeridas para una transacción, que memorablemente deletrear la palabra ACID

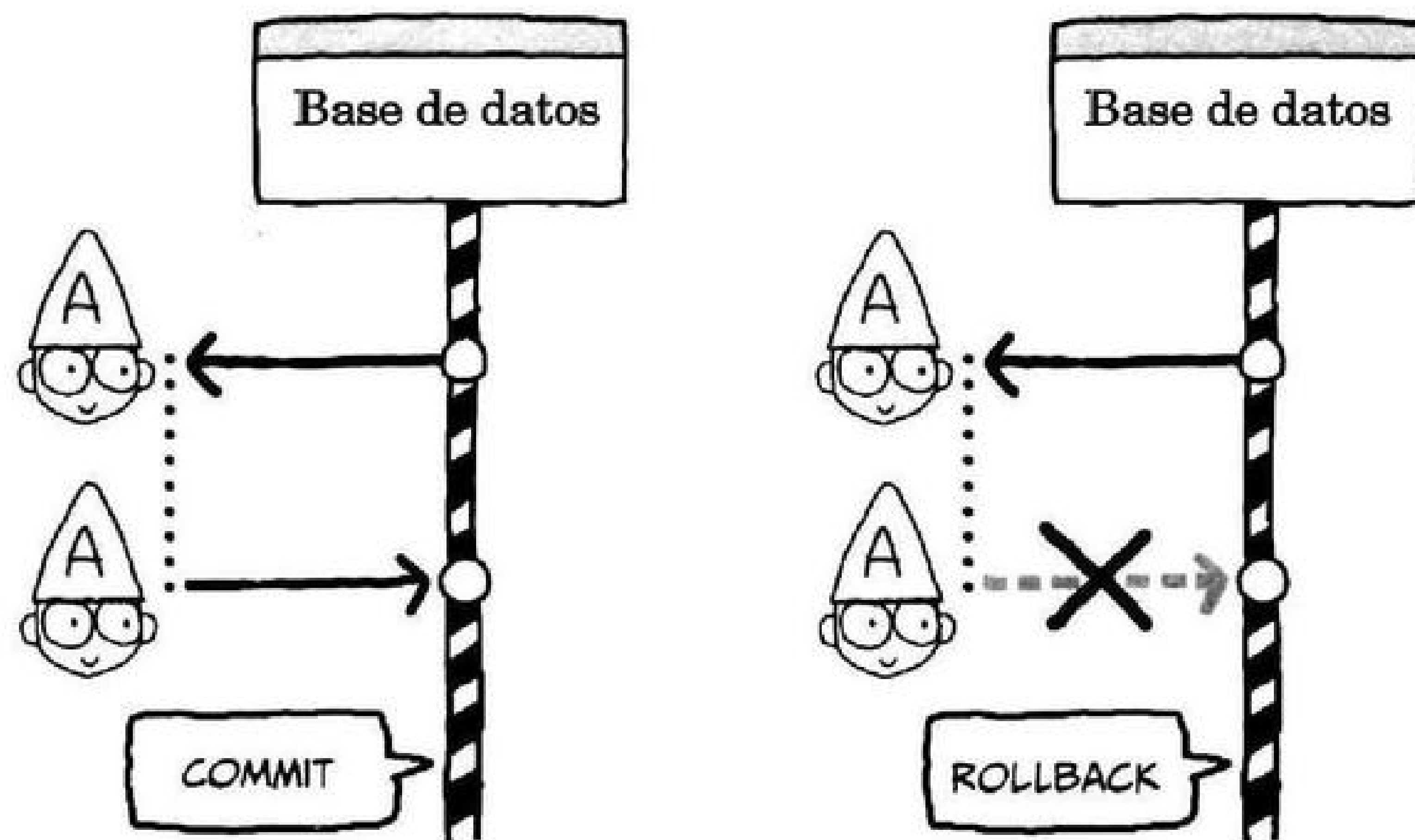
Propiedades requeridas para una transacción

Propiedad	Soporte de	Descripción
A	Atomicidad	Una transacción debe terminar con una operación de confirmación o retroacción.
C	Consistencia	Procesamiento de una transacción no resulta en la pérdida de consistencia de la base de datos.
I	Isolation (Aislamiento)	Incluso cuando las transacciones se procesan simultáneamente, los resultados deben ser los mismos que para el procesamiento secuencial.
D	Durabilidad	Los contenidos de una transacción completa no deberían verse afectado por fallo.

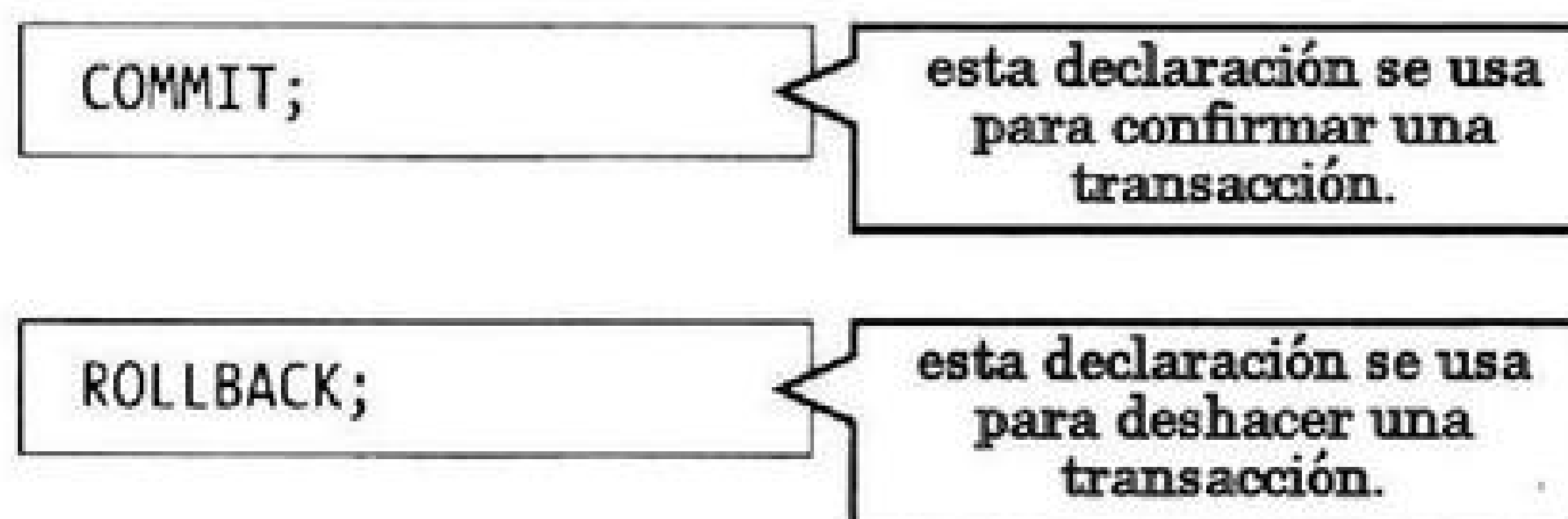
Examinemos cada una de estas propiedades en profundidad.

Atomicidad

La primera propiedad requerida para una transacción, atomicidad, significa que una transacción debe terminar ya sea con un commit o rollback con el fin de mantener una base de datos libre de inconsistencias. En resumen, ya sea toda la acción de una transacción se completan todas las acciones o se cancelan. Un commit finaliza la operación en la transacción. Un rollback cancela la operación en la transacción.



En algunos casos, una confirmación o retroacción se realiza automáticamente. También puede especificar que uno debe llevarse a cabo. Por ejemplo, puede especificar un retroceso si se produce un error. Puede utilizar SQL COMMIT y ROLLBACK para realizar estas operaciones.



PREGUNTAS

Responde a las siguientes preguntas para ver qué tan bien se entiende la atomicidad. Las respuestas están en la página 167.

Q1

Escriba una instrucción SQL que se puede utilizar para finalizar la transacción.

Q2

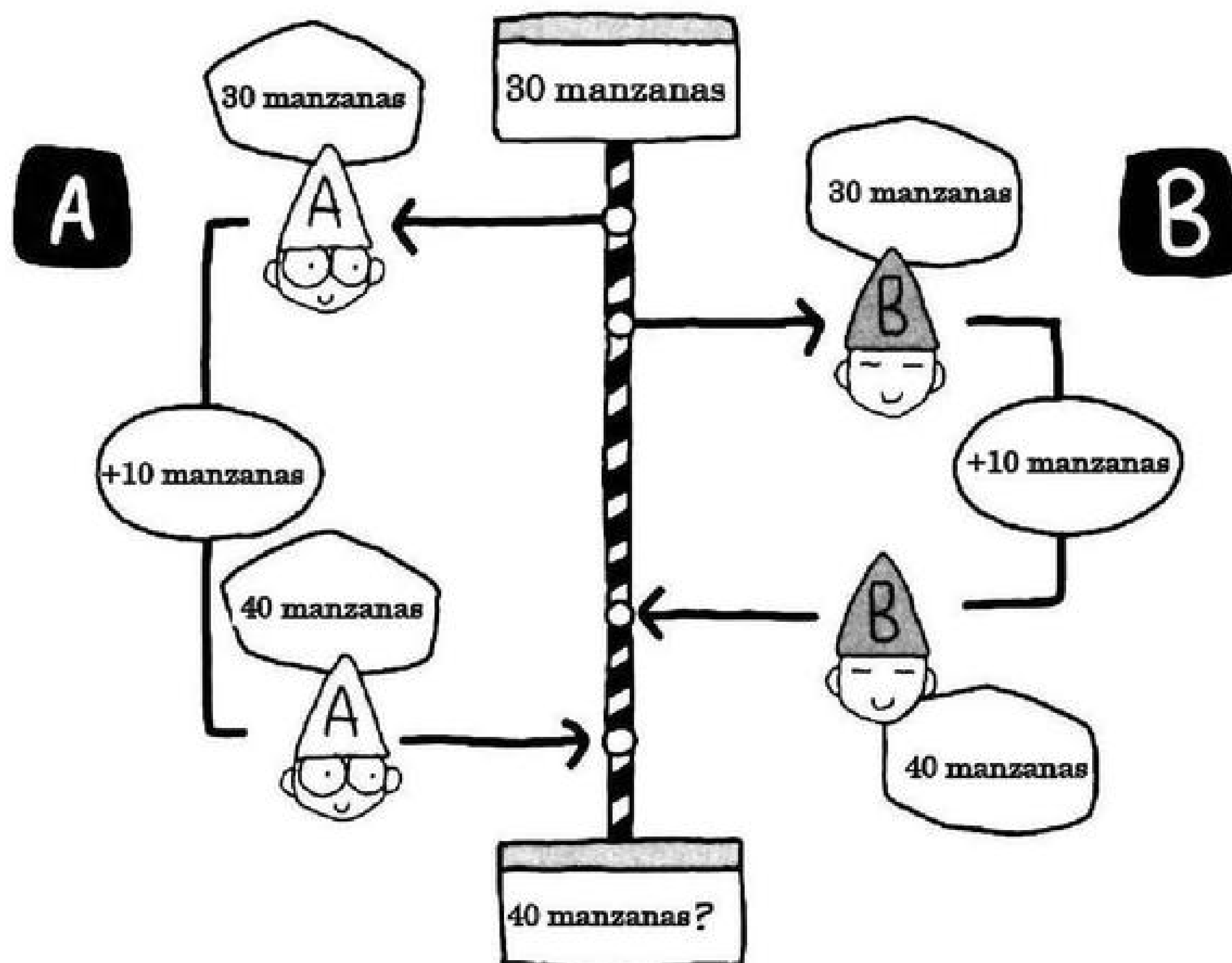
Escriba una instrucción SQL que se puede utilizar para cancelar una transacción.

CONSISTENCIA

Una transacción no debe crear errores. Si la base de datos es coherente antes de que una transacción sea procesada, entonces la base de datos también debe ser coherente después de que la transacción se produce.

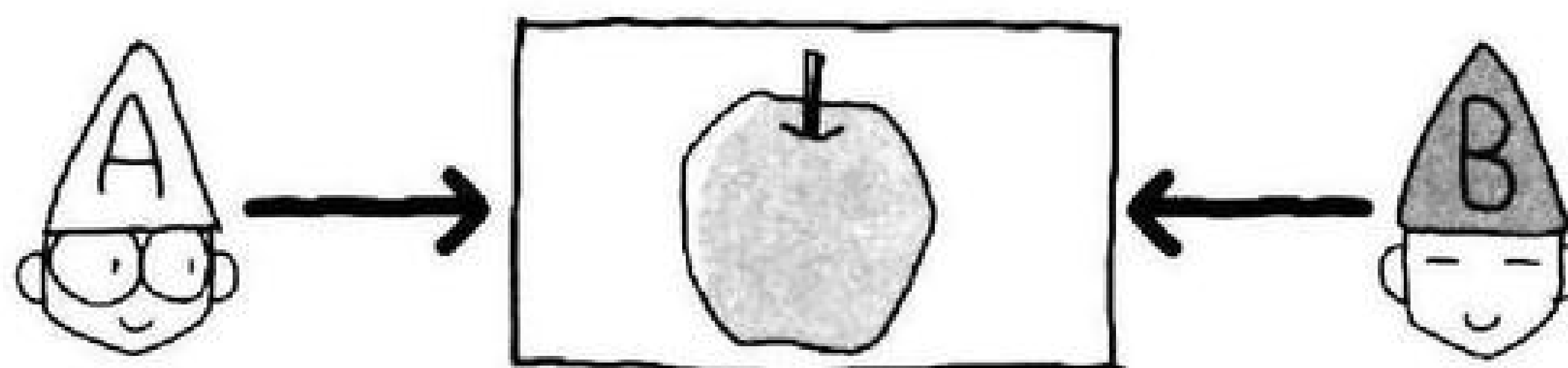
Calin ofreció el ejemplo de Andy y Becky cada uno tratando de añadir 10 manzanas con un total original de 30 manzanas. En lugar de obtención de la cantidad correcta de 50 manzanas, la base de datos muestra un total de 40 manzanas. Este tipo de error se llama una última actualización.





Cuando las transacciones se procesan simultáneamente, más de una transacción puede acceder a la misma tabla o fila al mismo tiempo y el conflicto de datos puede ocurrir.

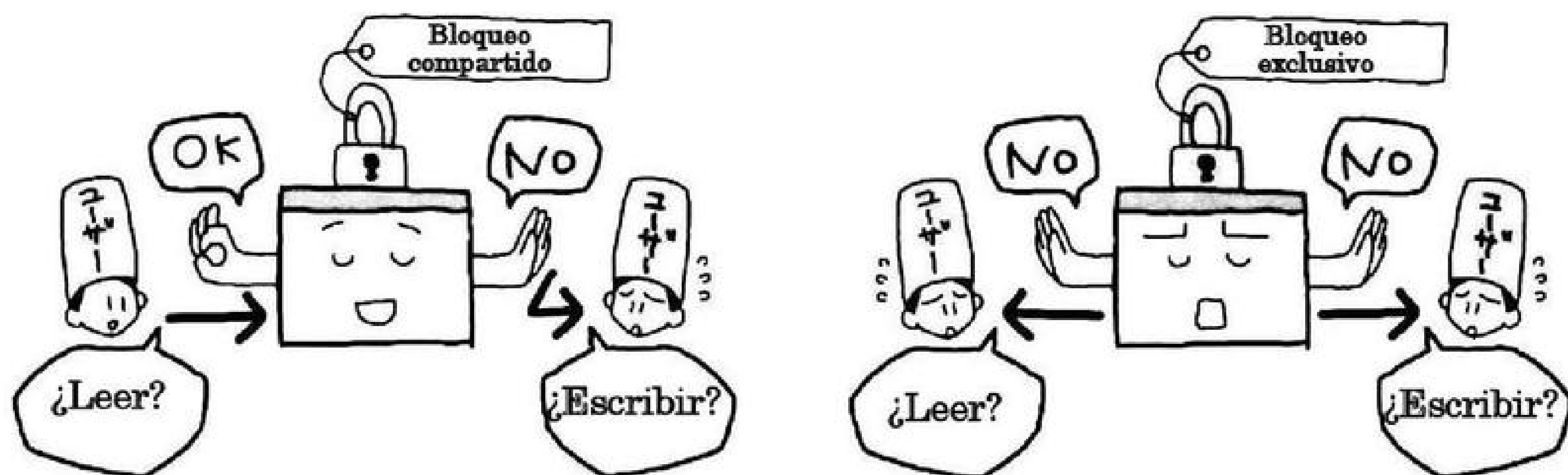
Tablas y filas sujetas a operaciones de una transacción se denominan recursos. En una base de datos, las transacciones deben ser capaces de acceder al mismo recurso al mismo tiempo sin crear inconsistencias.



ISOLATION (AISLAMIENTO)

Cuando dos o más transacciones simultáneas proporcionando el mismo resultado que si se realizaron en momentos separados, que el orden de procesamiento se prefiere como serializable. La propiedad de aislamiento requiere el programa para ser serializable y protege frente a errores.

Con el fin de hacer el pedido de procesamiento serializable, usted necesita tener el control sobre las operaciones que se intentan al mismo tiempo. El método más comúnmente usado para este propósito es el bloqueo basado en el control. Un bloqueo compartido se utiliza para leer los datos, mientras que un bloqueo exclusivo se utiliza para escribir datos.



Cuando un bloqueo compartido está en uso, otro usuario puede aplicar un bloqueo compartido para las otras operaciones, pero no un bloqueo exclusivo. Cuando un bloqueo exclusivo se aplica, otro usuario puede solicitar ya sea un bloqueo compartido o un bloqueo exclusivo para otras transacciones. A continuación se resume la relación entre un bloqueo compartido y el bloqueo exclusivo.

RELACIÓN DE COEXISTENCIA ENTRE TIPOS DE BLOQUEO

	Bloqueo compartido	Bloqueo exclusivo
Bloqueo compartido	Si	NO
Bloqueo exclusivo	NO	NO

PREGUNTAS

¿Entiendes el bloqueo? Contesta estas preguntas y verifique sus respuestas en la página 167

Q3

Cuando Andy se ha aplicado un bloqueo compartido, ¿puede Becky aplicar un bloqueo compartido?

Q4

Cuando Andy se ha aplicado un bloqueo exclusivo, ¿puede Becky aplicar un bloqueo compartido?

Q5

Cuando Andy se ha aplicado un bloqueo compartido, ¿puede Becky aplicar un bloqueo exclusivo?

Q6

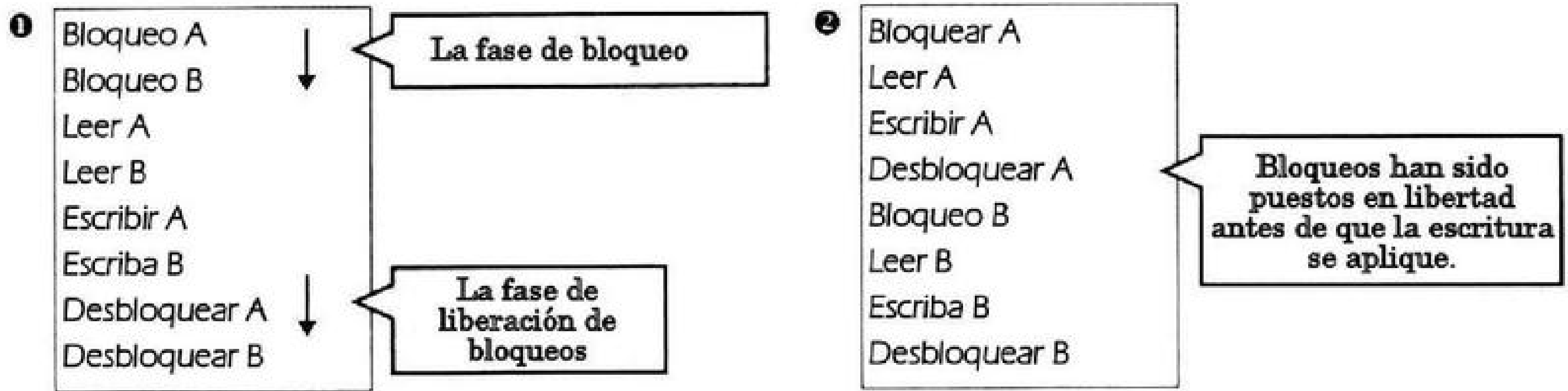
Cuando Andy se ha aplicado un bloqueo exclusivo, ¿puede Becky aplicar un bloqueo exclusivo?

BLOQUEO DE DOS FASES

En otras palabras, para asegurarse de que un programa es serializable, tenemos que obedecer las reglas específicas para el establecimiento y la liberación de los bloqueos. Una de estas reglas es bloqueo de dos fases, para cada operación, las dos fases se debe utilizar: un ajuste de bloqueos y otro para liberarlos.

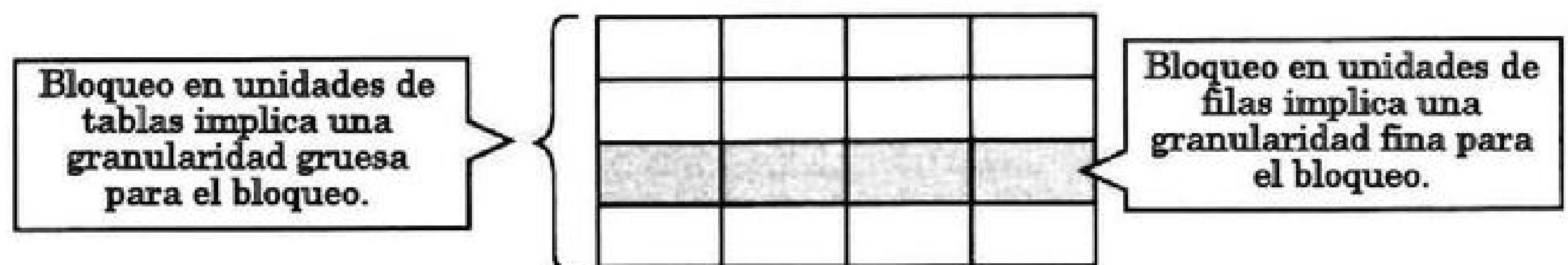
Por ejemplo, supongamos que hay recursos A y B, ambos de bloqueo. Transacción

❶ observa el estado de bloqueo de dos fases, mientras transacción ❷ no lo hace. La serialización sólo se puede conseguir cuando cada transacción compila con la regla de bloqueo de dos fases.



LA GRANULARIDAD DE BLOQUEO

Hay un número de recursos que se pueden bloquear. Por ejemplo, puede bloquear los datos en unidades de tablas o unidades de filas. El extendido en que los recursos están bloqueados se conoce como la granularidad. Granularidad gruesa ocurre cuando muchos recursos están bloqueados a la vez, y granularidad fina produce cuando pocos recursos están bloqueados.



Cuando granularidad es gruesa (o alta), el número de bloqueos necesarios por transacción se reduce, por lo que es más fácil de manejar granularidad. A su vez, esto reduce la cantidad de procesamiento requerido por la CPU en la que la base de datos está en funcionamiento. Por otro lado, a medida que más recursos están bloqueados, tiende a tomar más tiempo para esperar a que bloqueos para ser liberados utilizados por otras transacciones. Así, el número de transacciones que puede realizar tiende a disminuir cuando la granularidad es alta.

En contraste, cuando granularidad está bien (o menor), un mayor número de bloqueos se utilizan en una transacción, lo que resulta en más operaciones de gestión de bloqueos. Esto se traduce en una mayor procesamiento es exigido por la CPU. Sin embargo, ya menos recursos están bloqueados, usted pasará menos tiempo de espera para otras transacciones para liberar el bloqueo. Así, el número de transacciones que puede realizar tiende a aumentar.

PREGUNTAS

Responda a estas preguntas, y marque la respuesta correcta en la página 168.

Q7

El recurso de destino para el bloqueo ha sido cambiado de una tabla a fila. ¿Qué va a pasar con el número de transacciones que puede realizar al mismo tiempo?

Q8

El recurso de destino para el bloqueo se ha cambiado de una fila a una tabla. ¿Qué va a pasar con el número de transacciones que puede realizar al mismo tiempo?



OTROS CONTROLES DE CONCURRENCIA

Usted puede utilizar el bloqueo de llevar a cabo eficazmente dos o más operaciones al mismo tiempo. Sin embargo, utilizando el bloqueo viene con la carga de la gestión de bloqueo, interbloqueos desde los lugares donde las acciones del usuario conflictos pueden ocurrir. Métodos más sencillos para el control de concurrencia se puede utilizar cuando se tiene un pequeño número de transacciones o un alto número de operaciones de lectura. En tales casos, los métodos pueden utilizarse los siguientes:

Sello de Control de tiempo

Una etiqueta que contiene el tiempo de acceso, que se refiere como una marca de tiempo, se asigna a los datos accedidos durante una transacción. Si otra transacción con una marca de hora más tarde ya ha actualizado los datos, la operación no será permitida. Cuando una operación de lectura o escritura no está permitida, la transacción se revierte.

CONTROL DEL OPTIMISTA

Este método permite una operación de lectura. Cuando una operación de escritura se intenta, los datos se comprueban para ver si cualesquiera otras transacciones se han producido. Si otra transacción ya ha actualizado los datos, la transacción se revierte.

NIVELES DE AISLAMIENTO

En una base de datos del mundo real, se puede establecer el nivel de transacciones que se pueden procesar simultáneamente. Esto se conoce como nivel de aislamiento.

En SQL, la sentencia SET TRANSACTION se puede utilizar para especificar el nivel de aislamiento de las siguientes operaciones:

- LEER UNCOMMITTED
- LEER COMMITTED
- LECTURA REPETIBLE
- SERIALIZABLE

NIVEL DE AISLAMIENTO CONJUNTO TRANSACCIÓN DE LECTURA UNCOMMITTED;
--

Depending on the isolation level setting, any of the following actions may occur.

	Lectura sucia	Lectura no repetible	Lectura fantasma
LEER UNCOMMITTED	Possible	Possible	Possible
LEER COMMITTED	no se producirá	Possible	Possible
LECTURA REPETIBLE	no se producirá	no se producirá	Possible
SERIALIZABLE	no se producirá	no se producirá	no se producirá

- Una lectura sucia se produce cuando la transacción 2 lee una fila antes de la transacción 1 se ha comprometido.
- Una lectura no repetible se produce cuando una transacción lee los mismos datos dos veces y obtiene un valor diferente.
- Una lectura fantasma se produce cuando una transacción búsquedas para las filas que coincidan con una determinada condición, pero considera las filas erróneas debidas a los cambios de otra transacción.

DURABILIDAD

Una base de datos gestiona los datos importantes para garantizar la seguridad y la durabilidad en el caso de fracaso es crítica. La seguridad es también importante para prevenir el usuario no autorizado de escritura de datos y genera inconsistencias.

En una base de datos, se puede establecer el permiso para que pueda acceder a la base de datos o las tablas en él. Calin evita peligros a la base de datos del Reino mediante la mejora de la seguridad de la base de datos. En una base de datos relacional, la sentencia GRANT se utiliza para conceder permisos de lectura y escritura a los usuarios. Puede usar comandos GRANT para otorgar permiso a otros usuarios para la tabla de procesos que ha creado. Configuración de permisos es una tarea importante para el funcionamiento de bases de datos.

```
GRANT SELECT, UPDATE ON producto TO DEPARTAMENTO_NEGOCIOS_EXTRANJERO;
```

La sentencia Grant otorga el permiso para procesar los datos.

Puede asignar los siguientes privilegios (permisos) con sentencias SQL

PRIVILEGIOS DE BASES DE DATOS

Sentencia	Resultado
SELECT	Permite al usuario buscar filas en una tabla.
INSERT	Permite al usuario insertar filas en una tabla.
UPDATE	Permite al usuario actualizar filas de una tabla.
DELETE	Permite al usuario eliminar filas en una tabla.
ALL	Da todos los privilegios de usuario.

La concesión de un privilegio con WITH GRANT OPTION permite que el usuario pueda otorgar privilegios a otros usuarios. Con la declaración se muestra a continuación, el DEPARTAMENTO_NEGOCIOS_EXTRANJERO puede permitir a otros usuarios buscar y actualizar la base de datos.

```
GRANT SELECT, UPDATE ON producto TO DEPARTAMENTO_NEGOCIOS_EXTRANJERO;  
WITH GRANT OPTION;
```

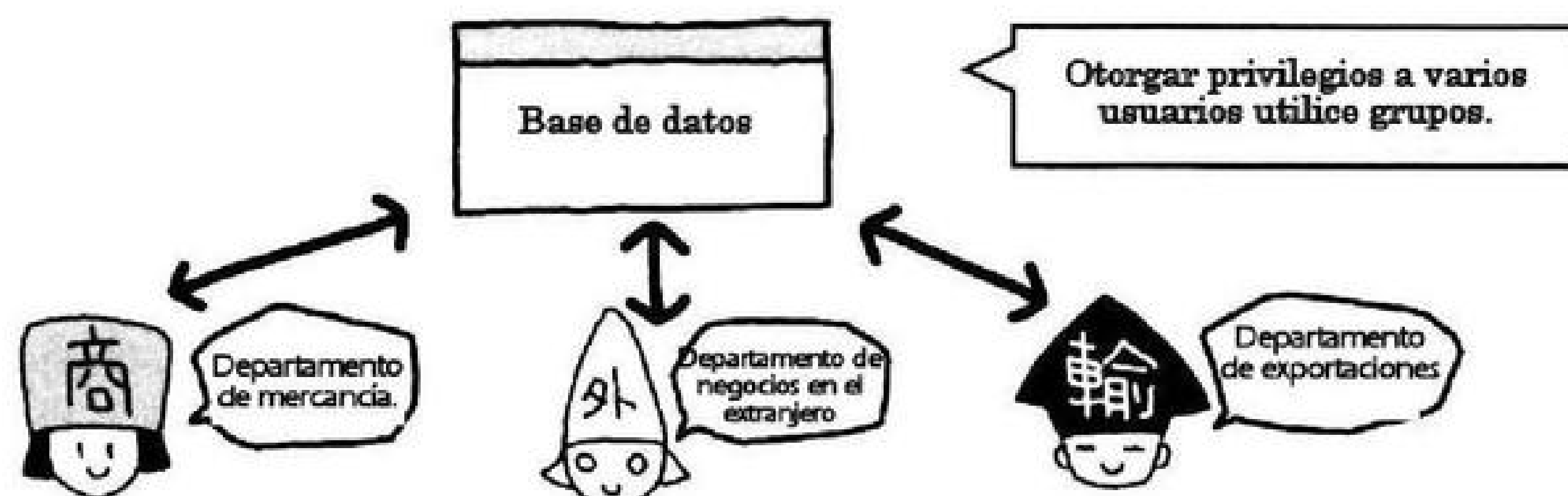
El usuario puede otorgar privilegios concedido a otros usuarios.

También puede quitar los privilegios de un usuario. Para ello, utilice la sentencia REVOKE.

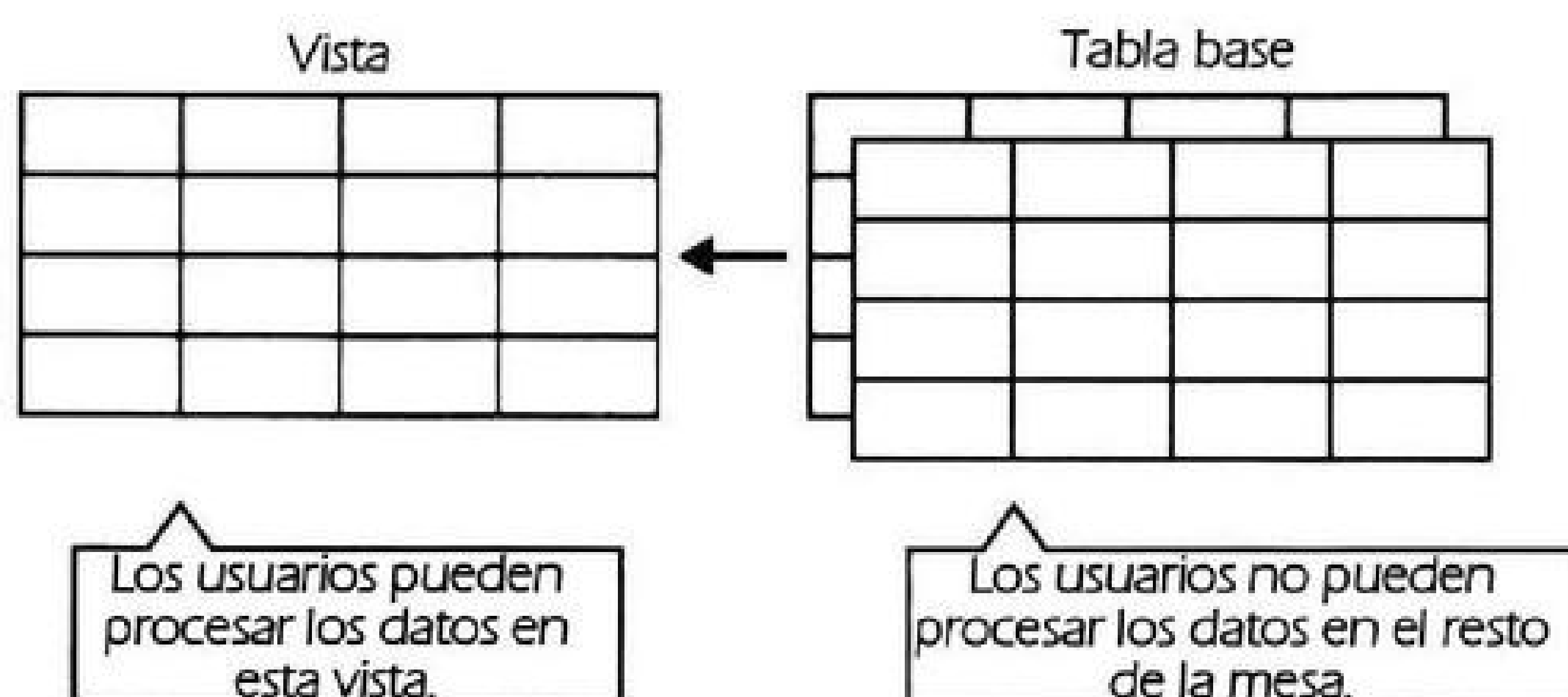
```
REVOKE SELECT, UPDATE ON producto FROM  
DEPARTAMENTO_NEGOCIOS_EXTRANJERO;
```

Esta sentencia revoca los privilegios del usuario.

Algunos de los productos de base de datos pueden agrupar una serie de privilegios y les conceda a varios usuarios a la vez. Agrupación hace más fácil la gestión de privilegios.



Uso de vistas, como se describe en la página 117, permite la gestión aún más controlada para mayor seguridad. En primer lugar, extraer parte de una tabla base para crear una vista. Configuración de un privilegio para este punto de vista significa el privilegio también se establece en la porción seleccionada de datos en la vista.



PREGUNTAS

Prueba estas preguntas en la durabilidad. Las respuestas están en la página 168.

Q9

Escriba una instrucción SQL que permite al departamento de exportación para buscar datos en la tabla de productos.

Q10

Cree una instrucción SQL para revocar el privilegio para eliminar datos de la tabla Producto.

Q11

Los privilegios se establecieron de la siguiente manera en una tabla de productos creados por el administrador. Introduzca un SÍ o NO en cada casilla de la tabla de abajo para indicar la presencia o ausencia del privilegio de cada departamento, respectivamente.

```
GRANT ALL producto TO Departamento_negocios_extranjero ;
GRANT INSERT, DELETE ON producto TO Departamento_de_mercancía ;
GRANT UPDATE, DELETE ON producto TO Departamento_de_exportaciones ;
```

	Search	Insert	Update	Delete
Departamento negocios extranjero				
Departamento de mercancía				
Departamento de exportaciones				

CUANDO OCURRE UN DESASTRE



Una base de datos debe tener un mecanismo que puede proteger los datos en el sistema en el caso de un fallo. Para garantizar la durabilidad de las transacciones, es obligatorio que ningún fallo puede crear datos incorrectos o defectuosos. Para protegerse de fracaso, una base de datos realiza diversas operaciones, que incluyen la creación de copias de respaldo y registros de transacciones.

TIPOS DE FALLAS

Insuficiencia de base de datos puede ocurrir en diversas circunstancias. Los posibles tipos de fallo son las siguientes tipos:

- Fallo de transacción
- Fallo del sistema
- Fallos medios

Fracaso de transacciones se produce cuando una transacción no se puede completar debido a un error en la propia transacción. La transacción se revierte cuando se produce este error.

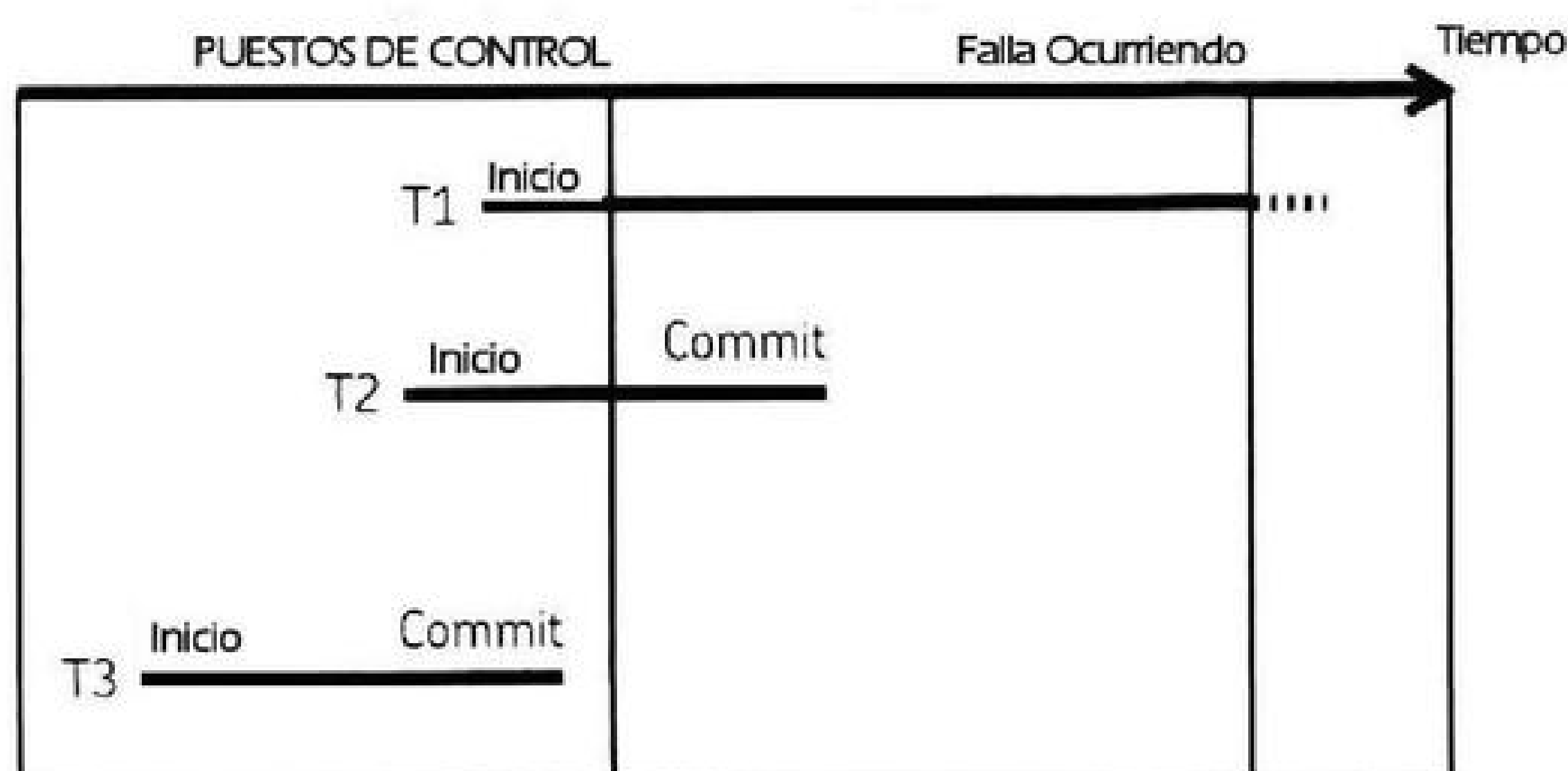
Fallo del sistema se produce cuando el sistema se apaga debido a una falla en la fuente de este tipo. En el caso de un sistema de recuperación de desastres se lleva a cabo después de que se reinicie el sistema. En general, las transacciones que aún no han sido confirmadas en el momento de la falla se deshacen, y los que ya se han comprometido en caso de avería se ruedan hacia adelante.

Fallo de los medios se produce cuando el disco duro que contiene la base de datos está dañado. En el caso de un fallo de los medios de comunicación, recuperación de desastres se lleva a cabo utilizando los archivos de copia de seguridad. Transacciones con posterioridad a los archivos de copia de seguridad se crean en avance.

PUESTOS DE CONTROL

Con el fin de mejorar la eficiencia de una operación de escritura en una base de datos, un **tampón** (un segmento de memoria utilizada para almacenar temporalmente los datos) se utiliza a menudo para escribir datos en el corto plazo. El contenido de la memoria intermedia y la base de datos están sincronizados y, a continuación un punto de control está escrito.

Cuando la base de datos, escribe un puesto de control, no tiene que realizar una recuperación de fallos para las transacciones que se cometieron antes del puesto de control. Las transacciones que no fueron cometidos antes del puesto de control deben ser recuperadas. Supongamos ahora que las transacciones se muestran a continuación se llevan a cabo en el momento de un fallo en el sistema. ¿Qué transacciones se deshacen? ¿Cuáles deberían ser Roll Forward?





PREGUNTAS

Prueba estas preguntas en base a la tabla de la página anterior. Las respuestas están en la página 168.

Q12

¿Cómo debe ser procesado T1?

Q13

¿Cómo debe ser procesado T2?

Q14

¿Cómo debe ser procesado T3?

En caso de fallo de la base de datos, los mecanismos de recuperación descritos anteriormente proteger la base de datos contra la incoherencia. Es por eso que usted puede estar seguro de la integridad de la base de datos cuando se utiliza.

ÍNDICES

Una base de datos gestiona grandes cantidades de datos, por lo que la búsqueda de datos específicos puede tomar mucho tiempo. ¡Pero usted puede usar índices para acelerar las búsquedas!

Codigo producto	Nombre de producto	Precio unitario	Distrito
101	Melon	800G	Mar del Sur
102	Fresa	150G	Medio
103	Manzana	120G	Mar del Norte
104	Limon	200G	Mar del Sur
201	Castaña	100G	Mar del Norte
202	Caqui	160G	Medio
301	Melocoton	130G	Mar del Sur
302	Kiwi	200G	Mar del Sur

Se requiere mucho tiempo para buscar cada elemento fila por fila.

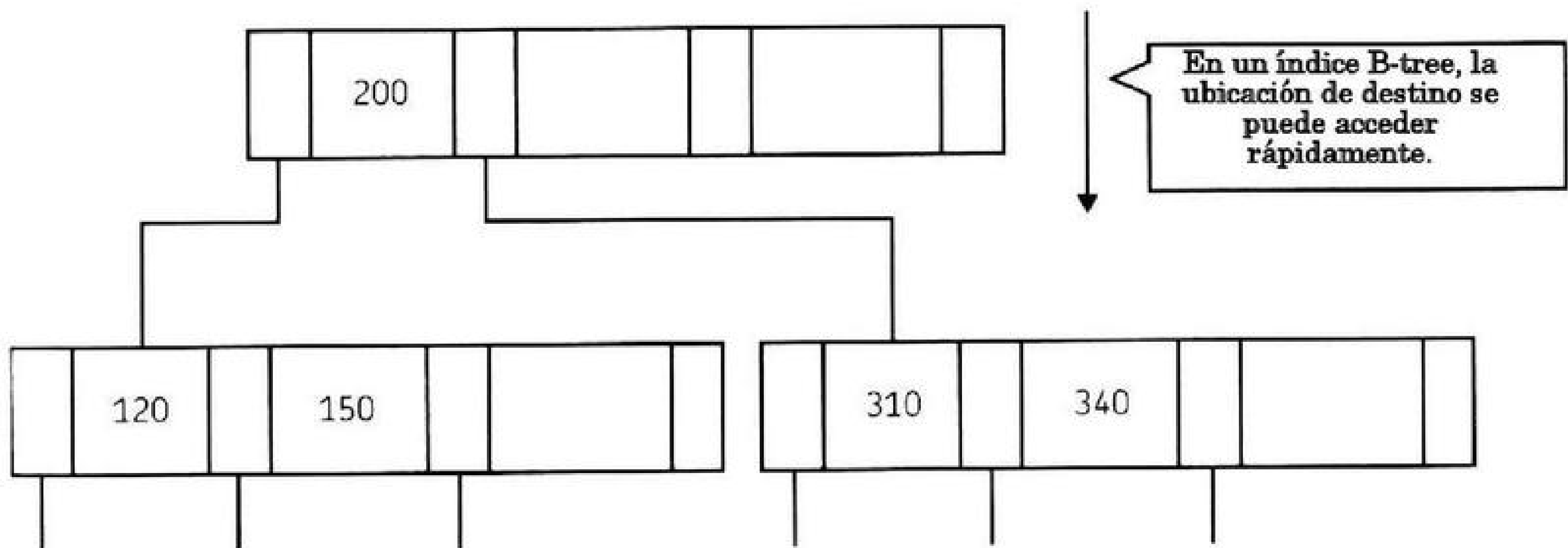
Un índice es una herramienta que le permite acceder rápidamente la ubicación de los datos de destino. Al buscar algunos datos en una base de datos grande, la búsqueda con índices promete resultados rápidos.

Codigo producto	Nombre de producto	Precio unitario	Distrito
101	Melon	800G	Mar del Sur
102	Fresa	150G	Medio
103	Manzana	120G	Mar del Norte
104	Limon	200G	Mar del Sur
201	Castaña	100G	Mar del Norte
202	Caqui	160G	Medio
301	Melocoton	130G	Mar del Sur
302	Kiwi	200G	Mar del Sur

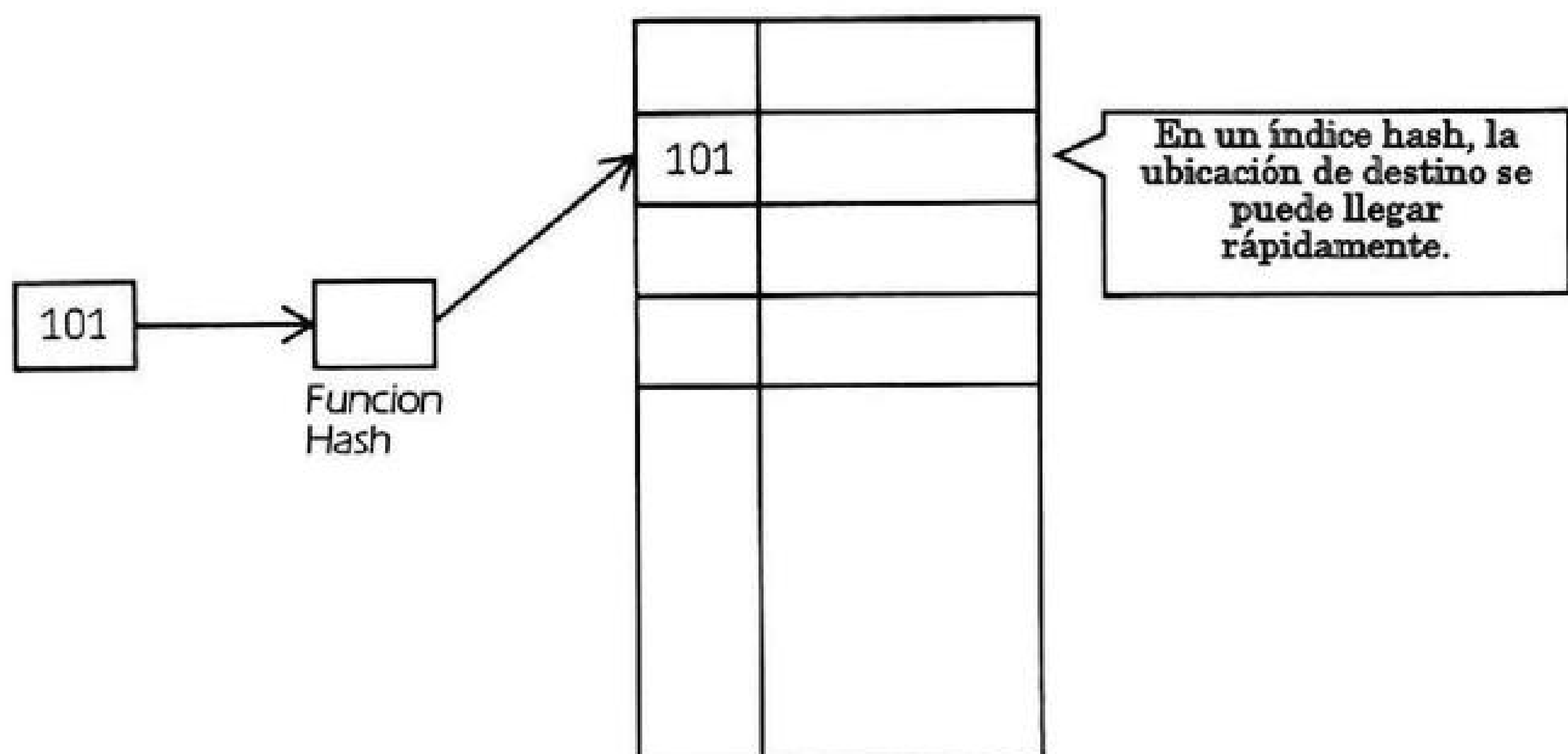
La ubicación de datos de destino se puede acceder rápidamente usando su índice.

índice

Métodos de indexación incluyen B-tree y métodos de hash se compone un índice B-tree de nodo principal y los nodos secundarios, que pueden tener más nodos secundarios. Los nodos se organizan de forma ordenada. Cada matriz contiene información acerca de los valores mínimos y máximo contenida por todos sus hijos. Esto permite que la base de datos para navegar rápidamente a la posición deseada, saltándose secciones enteras del árbol que no puede contener el valor deseado.



El método de Índice hash encuentra la ubicación de los datos del objetivo al aplicar una función hash a los valores de clave de los datos. Los actos de hash como una huella digital única para un valor. El método del Índice hash puede realizar determinadas búsquedas full-partido, como una búsqueda de código de producto 101. Sin embargo, no está diseñado para buscar eficazmente para condiciones comparativas, como los códigos de producto no menos de 101 o referencias difusas como los productos con nombres que terminan en n.



En algunos casos, el uso de un índice no puede acelerar la búsqueda que utilizan un índice no ahorrar tiempo a menos que usted está buscando sólo una pequeña parte de los datos. Además, hay casos en los índices se recrean todos los datos son actualizados, lo que resulta en el procesamiento más lento de una operación de actualización.



PREGUNTAS

Prueba estas preguntas en la indexación. Las respuestas están en la página 168.

Q15

¿Qué índice sería más potente en una búsqueda con un signo igual, un B-tree o índice hash?

Q16

¿Qué índice sería más poderoso en una búsqueda con un signo de desigualdad, un hash B-tree o índice hash?

OPTIMIZACIÓN DE UNA CONSULTA

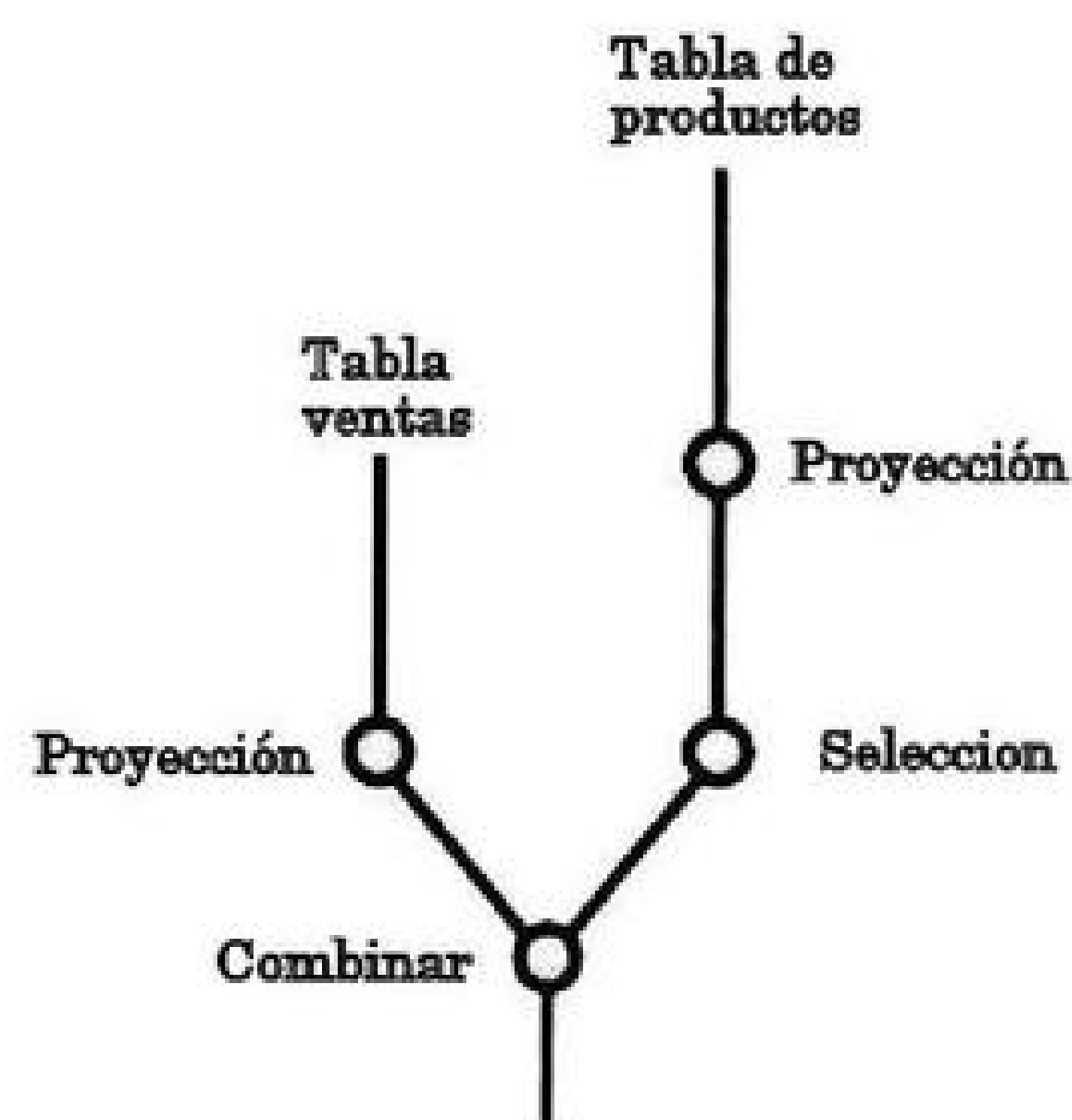
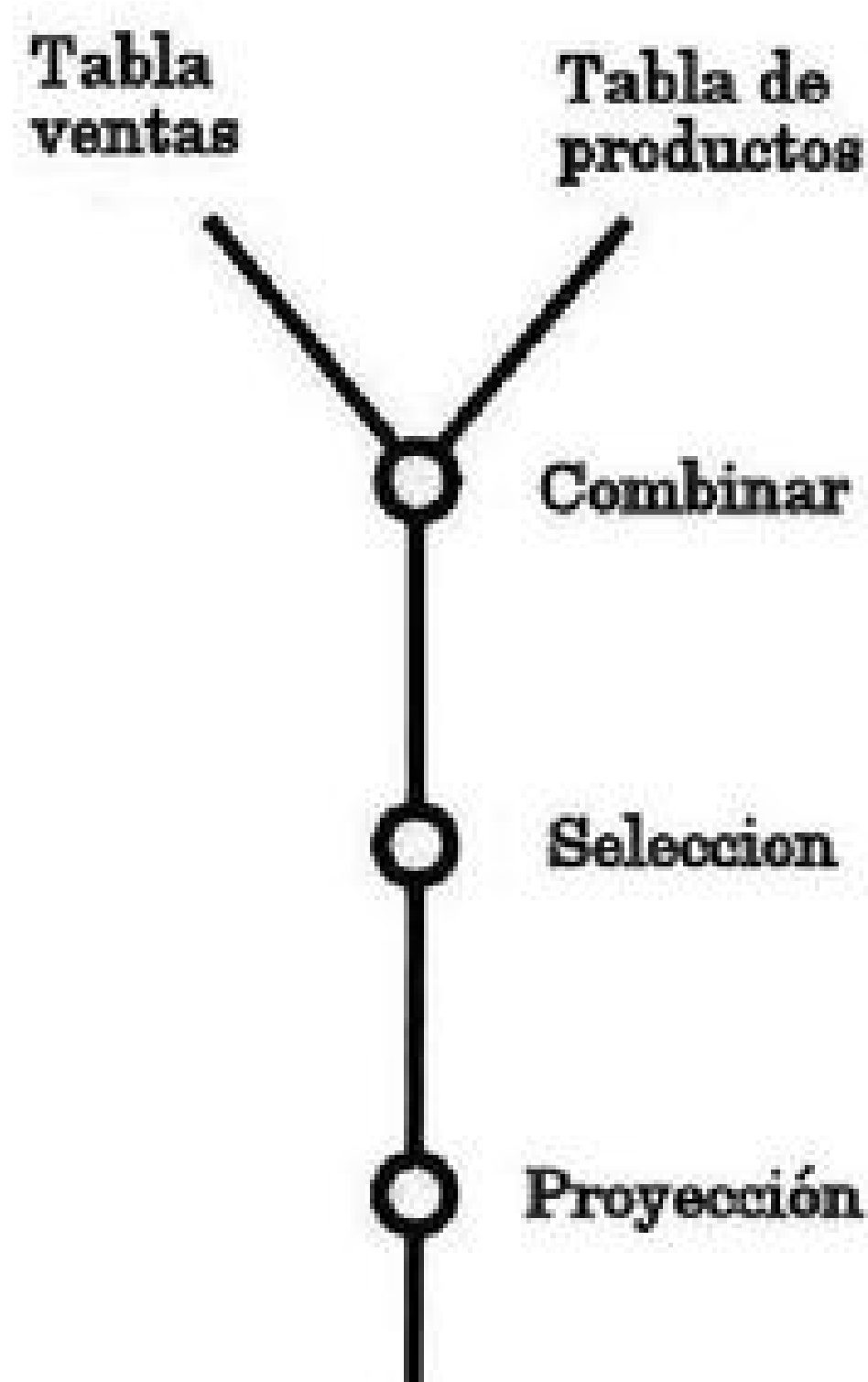
Cuando se consulta una base de datos, la base de datos analiza la consulta SQL y considera la posibilidad de utilizar un índice para que pueda procesar la consulta con mayor rapidez. Vamos a examinar el procedimiento para la tramitación de una consulta.

La base de datos puede decidir sobre un orden óptimo para procesar una consulta. La mayoría de consultas se pueden procesar en varios órdenes con los mismos resultados, pero con velocidades posiblemente diferentes. Por ejemplo, supongamos que hay una consulta para extraer fechas de venta y de productos para productos con un precio mayor que unen 200G. Esta consulta puede ser vista como que consiste en los pasos siguientes.

```
SELECT fecha, nombre_producto
FROM   producto, ventas
WHERE  precio_unitario >=200
AND    producto.codigo_producto = ventas.codigo_producto ;
```

1. Unir a la tabla de productos y la tabla de ventas.
2. Seleccione un producto cuyo precio unitario es superior a 200G.
3. Extraer columnas de fechas y nombres de productos.

Por ejemplo, la figura de la izquierda muestra la consulta procesada en orden de 1 a 3. La figura de la derecha muestra la consulta procesada en el orden de 3 a 1. De cualquier manera, las consultas son equivalentes.



Sin embargo, cuando se procesa 1-3, la misma consulta general requiere un tiempo de procesamiento, porque cuando la combinación se lleva a cabo primero, una tabla intermedia con muchas filas puede ser creada. Por otra parte, el procedimiento de 3 a 1 requiere un tiempo de procesamiento más corto, ya que la selección y la proyección ocurrir primero, el recorte de los datos no deseados tan pronto como sea posible. Por lo tanto, la misma consulta puede requerir un tiempo de procesamiento diferente, dependiendo del orden en que se llevan a cabo proyección, selección, y combinación.

En general, la base de datos debe utilizar las siguientes reglas para encontrar el mejor orden de consulta:

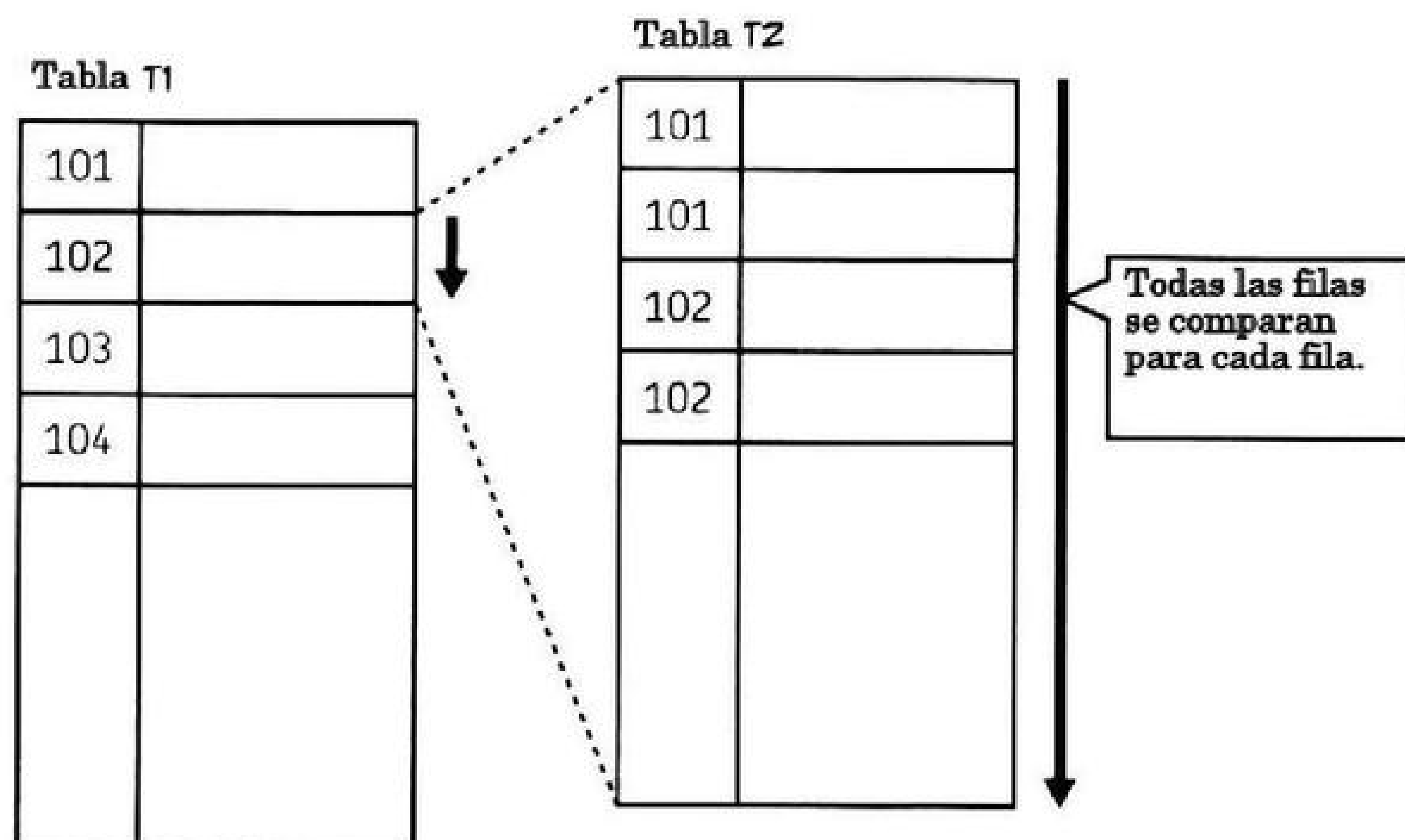
- Ejecutar selección primero para reducir el número de filas.
- Ejecutar primera proyección para reducir el número de columnas irrelevantes para el resultado.
- Ejecutar inscribirse más tarde.

Existen diferentes técnicas para la ejecución de proyección, selección y unirse, respectivamente.

Para la selección, se puede utilizar una búsqueda en todo el partido o una búsqueda basado en índices. En combinación, los métodos disponibles son los siguientes.

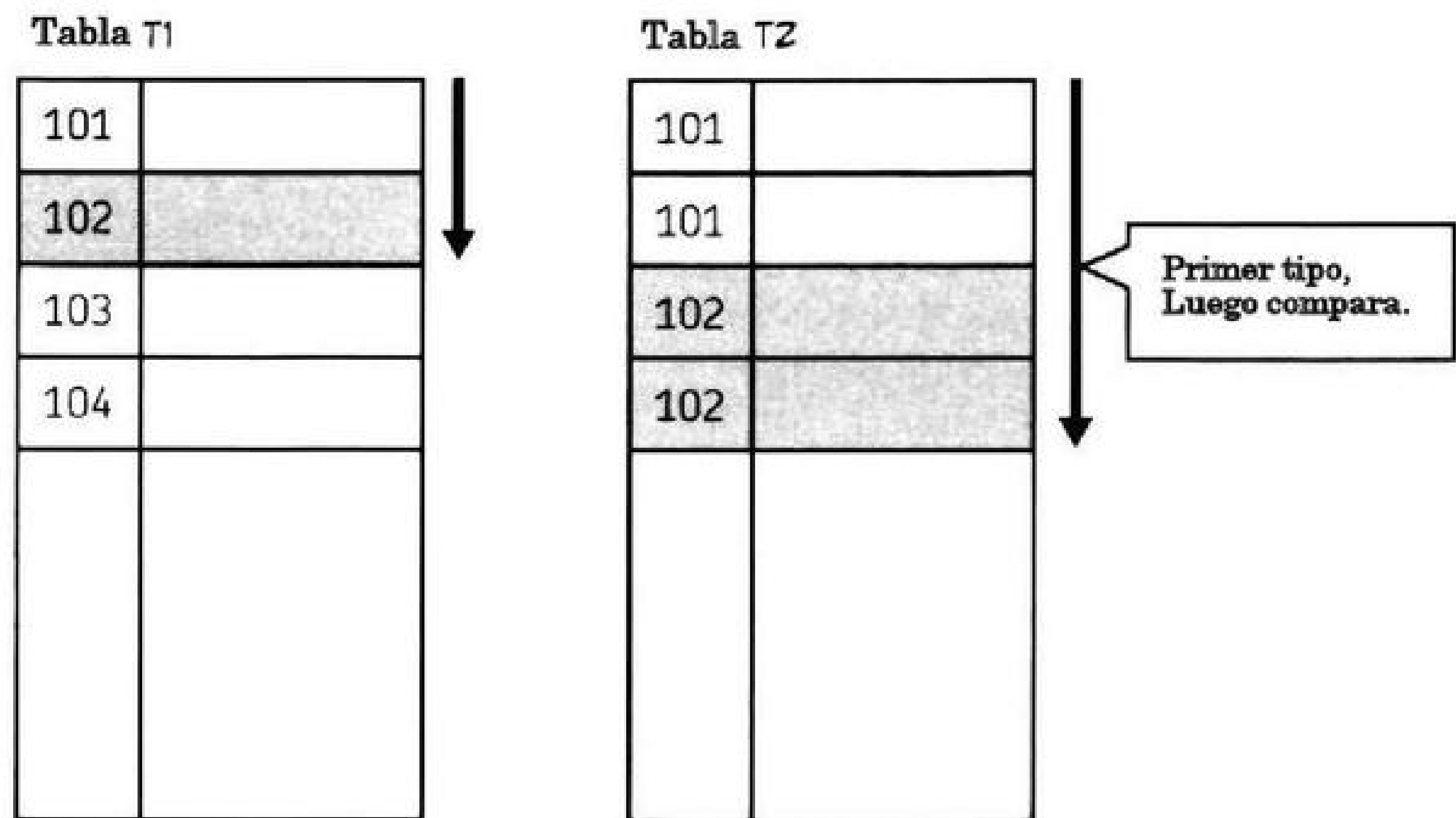
BUCLE ANIDADO

El método de bucle anidado compara una fila en una tabla de varias filas de otra tabla (ver la figura siguiente). Por ejemplo, uno de los valores de una fila de una tabla T1 se utiliza para encontrar las filas coincidentes en la tabla T2. Si los valores son iguales, entonces se unió a una fila se crea.



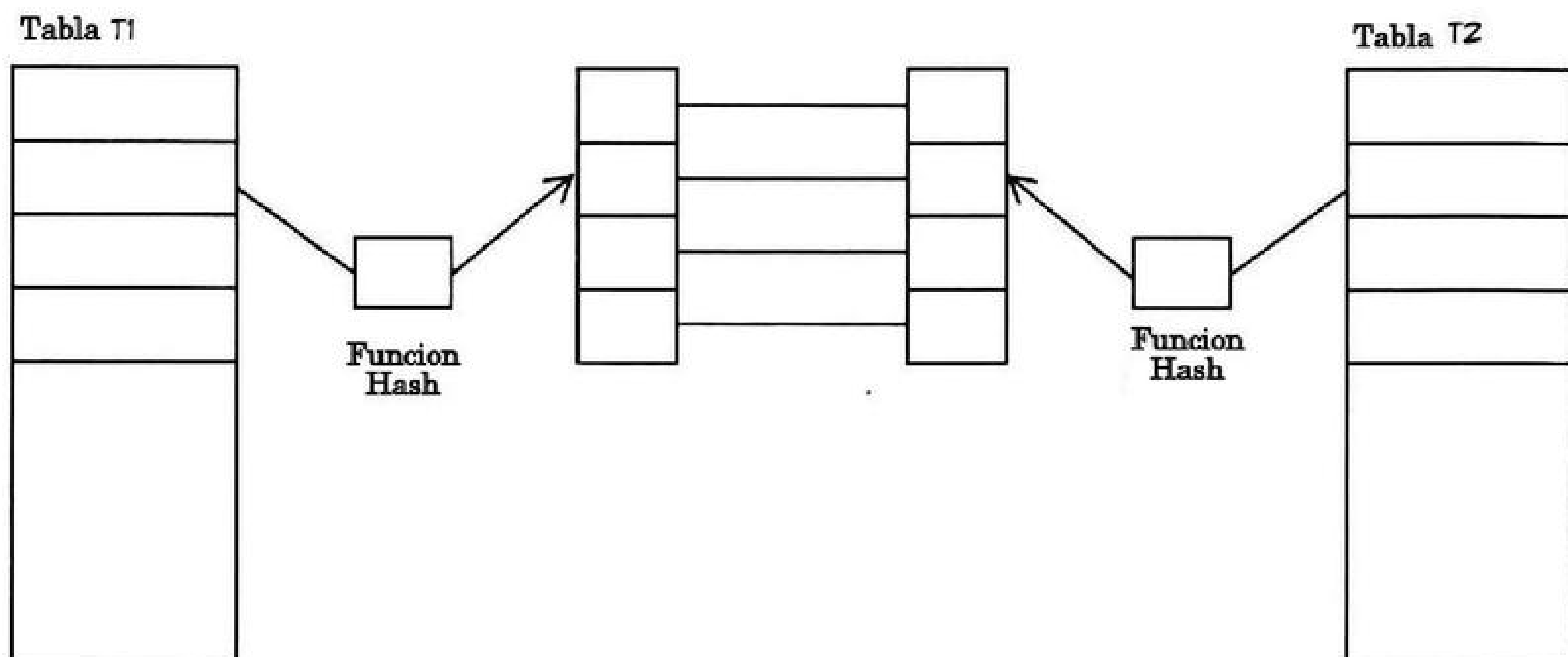
Tipo fusión

El método tipo fusión y la fusiona filas en varias tablas (ver la figura siguiente). En primer lugar, toda o parte de la tabla T1 y T2 están ordenadas. A continuación, se comparan a partir de la fila superior, y se unió a una fila cada vez que se crea el mismo valor que se encuentra. Puesto que ya se han ordenado, procesar sólo hay que hacer en una dirección, por lo que requerirá menos tiempo. Debe tener en cuenta, sin embargo, de las necesidades de tiempo para la clasificación inicial.



HASH

Un hash divide una de las tablas mediante una función hash y luego la fusiona con una fila de otra tabla que tiene el mismo valor hash. Este método eficaz selecciona la fila para inscribirse.



OPTIMIZADOR

Cuando se procesa una consulta, estas técnicas se examinan diferentes para un rendimiento óptimo en una base de datos, la función a cargo de la optimización de las consultas que se conoce como el optimizador. Hay dos tipos comunes.

PROCESAMIENTO BASADO EN NORMAS

Ciertas reglas se establecen antes de que se realicen las operaciones. Por ejemplo, algunas operaciones se pueden combinar o reordenan en mucho la misma manera puede ser una ecuación algebraica manipulado y aún significan la misma cosa. El optimizador trata de encontrar la manera más eficiente para procesar la consulta que da el mismo resultado.

PROCESAMIENTO BASADO EN COSTO

Este método trata de estimar el costo de procesamiento de la consulta, en base a las estadísticas que mantiene la base de datos. Costo de procesamiento basado a veces es más flexible que la regla basada en el procesamiento. Pero requiere actualizaciones periódicas de la base de datos estadística. La gestión y el análisis de estas estadísticas requieren mucho tiempo.

RESUMEN



- Puede configurar los privilegios de usuario para una base de datos.
- El bloqueo asegura la coherencia en una base de datos tiene usuarios múltiples.
- Indexación permitir búsquedas rápidas.
- Una base de datos con funciones de recuperación de desastres.

RESPUESTAS

Q1

COMMIT;

Q2

ROLLBACK;

Q3 Si

Q4 No

Q5 No

- Q6** No
- Q7** aumenta
- Q8** disminuye
- Q9**

GRANT SELECT ON producto TO Departamento_exportaciones ;

Q10

REVOKE DELETE ON producto FROM Departamento_Negocios_Extranjero ;

Q11

	Search	Insert	Update	Delete
Departamento negocios extranjero	Si	Si	Si	Si
Departamento de mercancía	NO	Si	NO	Si
Departamento de exportaciones	NO	NO	Si	Si

Q12

Un rollback se lleva a cabo ya que no se compromete en el momento de la ocurrencia fracaso.

Q13

Un roll forward se lleva a cabo ya que se ha comprometido en el momento de la ocurrencia fracaso.

Q14

Ninguna operación de recuperación que necesitaba ya que se ha comprometido en el momento del punto de control.

Q15

Hash

Q16

B-tree

6

¡BASES DE DATOS ESTÁN POR TODAS PARTES!





* OM NOM NOM!

は、は、は!!

¡Qué delicia! ¡Esta fruta es del Reino de Kod!

¡Padre!

NO, NO!

¿Sí? ¿Cuál es el problema?
¿Quieres un plátano, también, ¿Ruruna?

Padre comer fruta es todo lo que has hecho desde que has vuelto.

¡Perdóname!
¡Ninguna otra fruta se compara!

Pero lo admito, Ruruna has mantenido un estricto control mientras yo estaba fuera.

¡Mira cómo prospera el Reino de Kod!

En definitiva, una base de datos es ¡una cosa conveniente!

は、は、は

ホホホホ



Pero usted debe haber llegado por algún otro negocio, ¿no?

Sí, ¡es cierto!



T ... Tico ¿has visto a Tico?

TICO?



Sí. Oh... tico es una chica así de grande, y vuela en el aire...

¿De qué estás hablando? Nunca la había visto antes.

Salió del libro que me diste, padre.



Ella fue una gran ayuda para mí y Caín mientras usted no estaba.

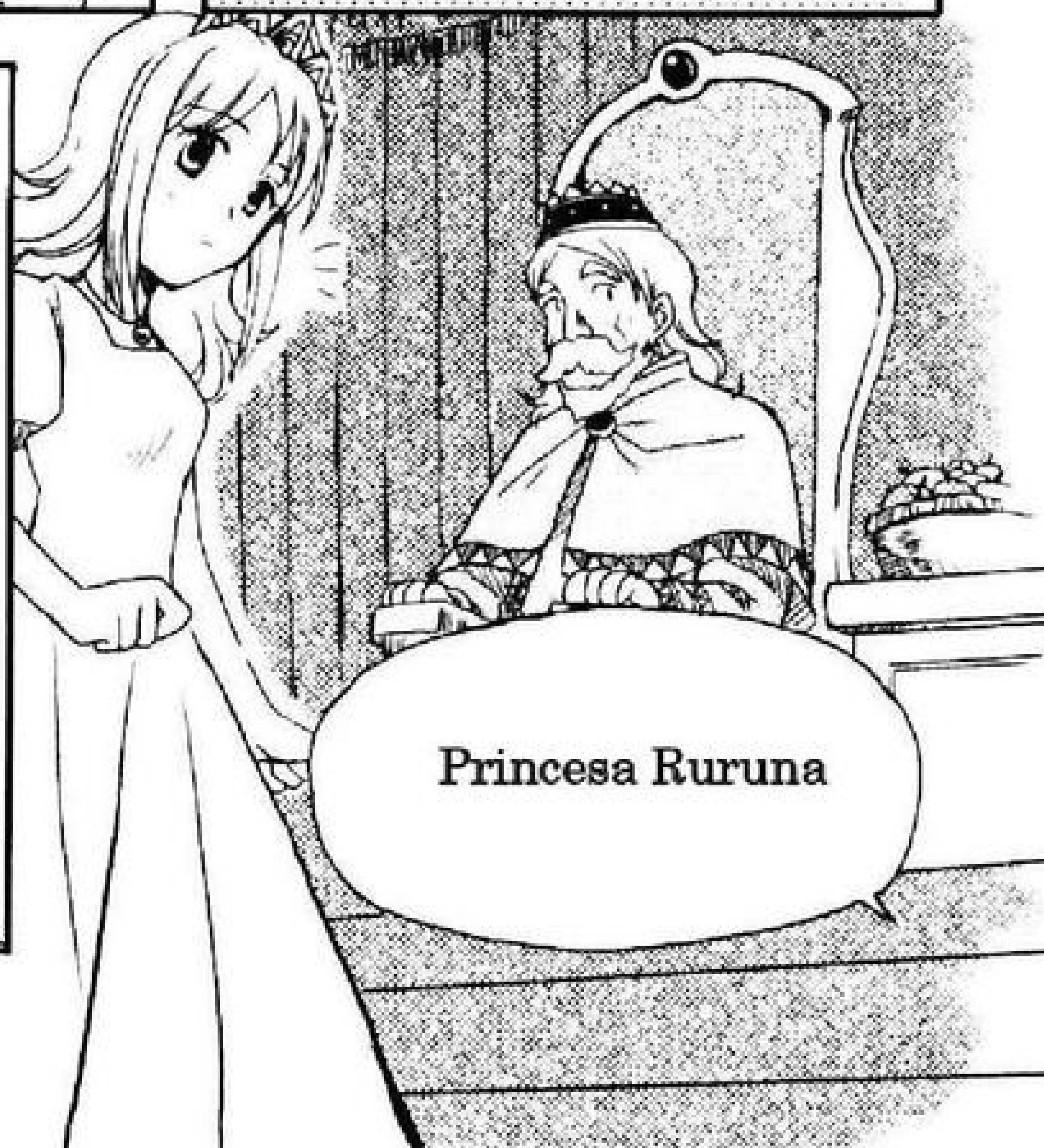


Pensé que mi padre la conocía porque ella salió de el libro que me dio.

¿La has visto?

No, no lo he hecho.

Mientras pensaba, tico es invisible excepto paraa Caín y para mí.



Princesa Ruruna

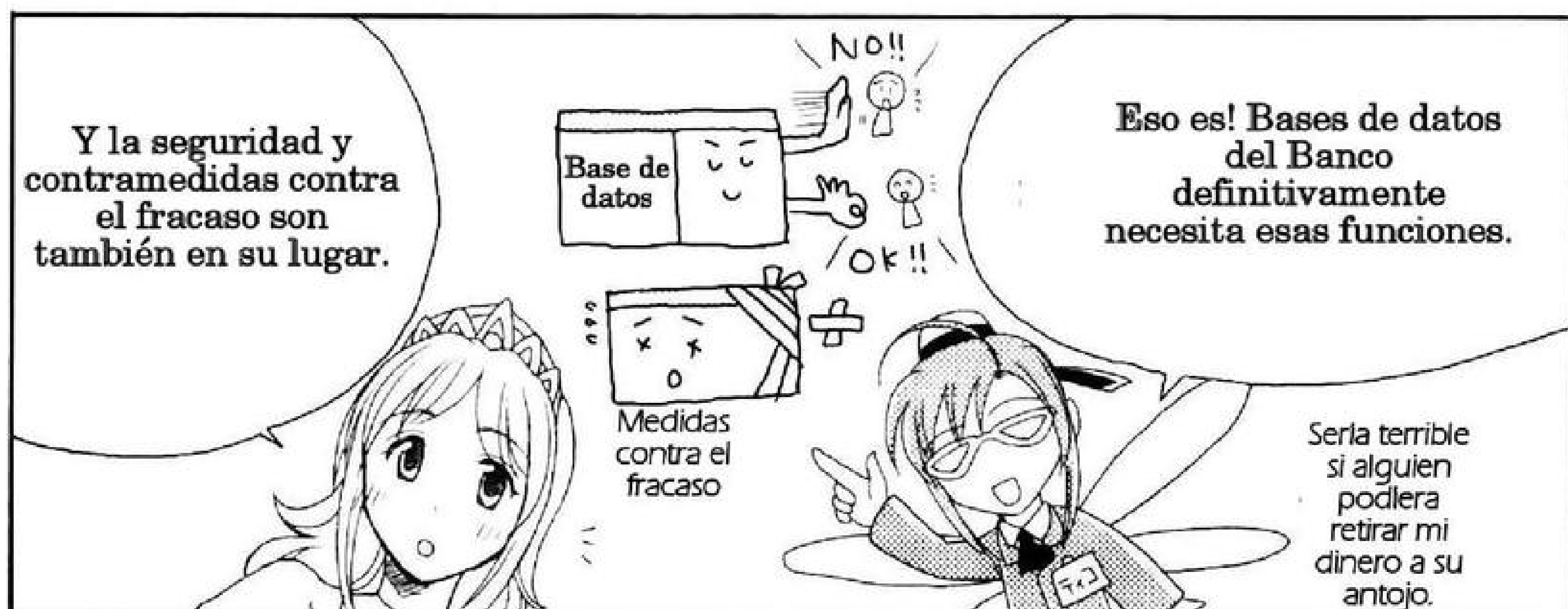




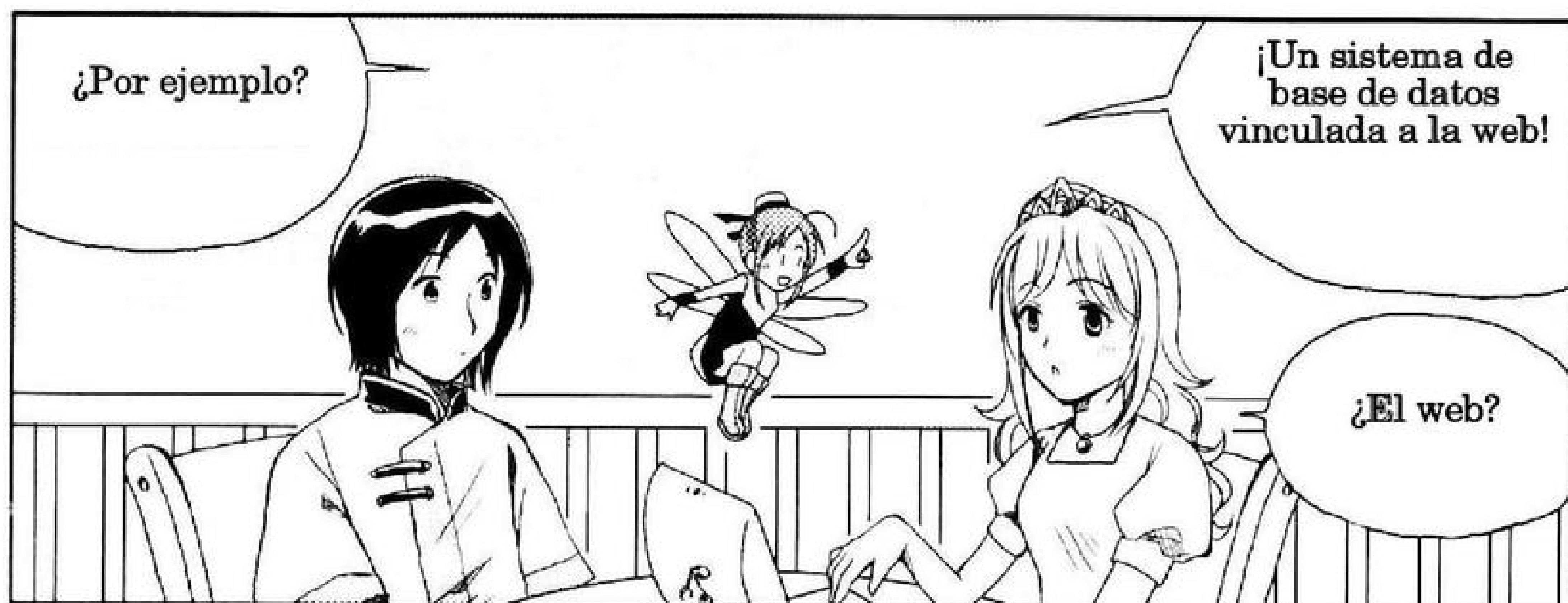


DATABASES IN USE





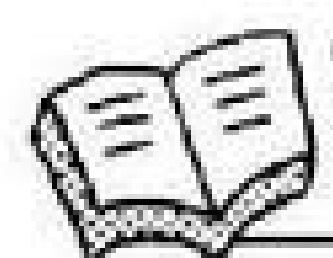
Base de datos y la web



Cuando
usted busca
un cierto
libro que
usted tiene
en mente,

ほん

Introduzca una
palabra clave en un
navegador web.



Book
search▶▶▶

CATEGORIES ▼

KEYWORD

TICO SEARCH



TICO
SEARCH

¿Qué clase de
libros es lo que
buscas?

Bueno, déjame ver... sería
"fruta", supongo.

A continuación,
escriba la fruta
en este campo
de palabras
clave.

KEYWORD

Fruit

Esta palabra clave se
envía como una solicitud
HTTP.

Un equipo que recibe
una solicitud y procesos
es un servidor.



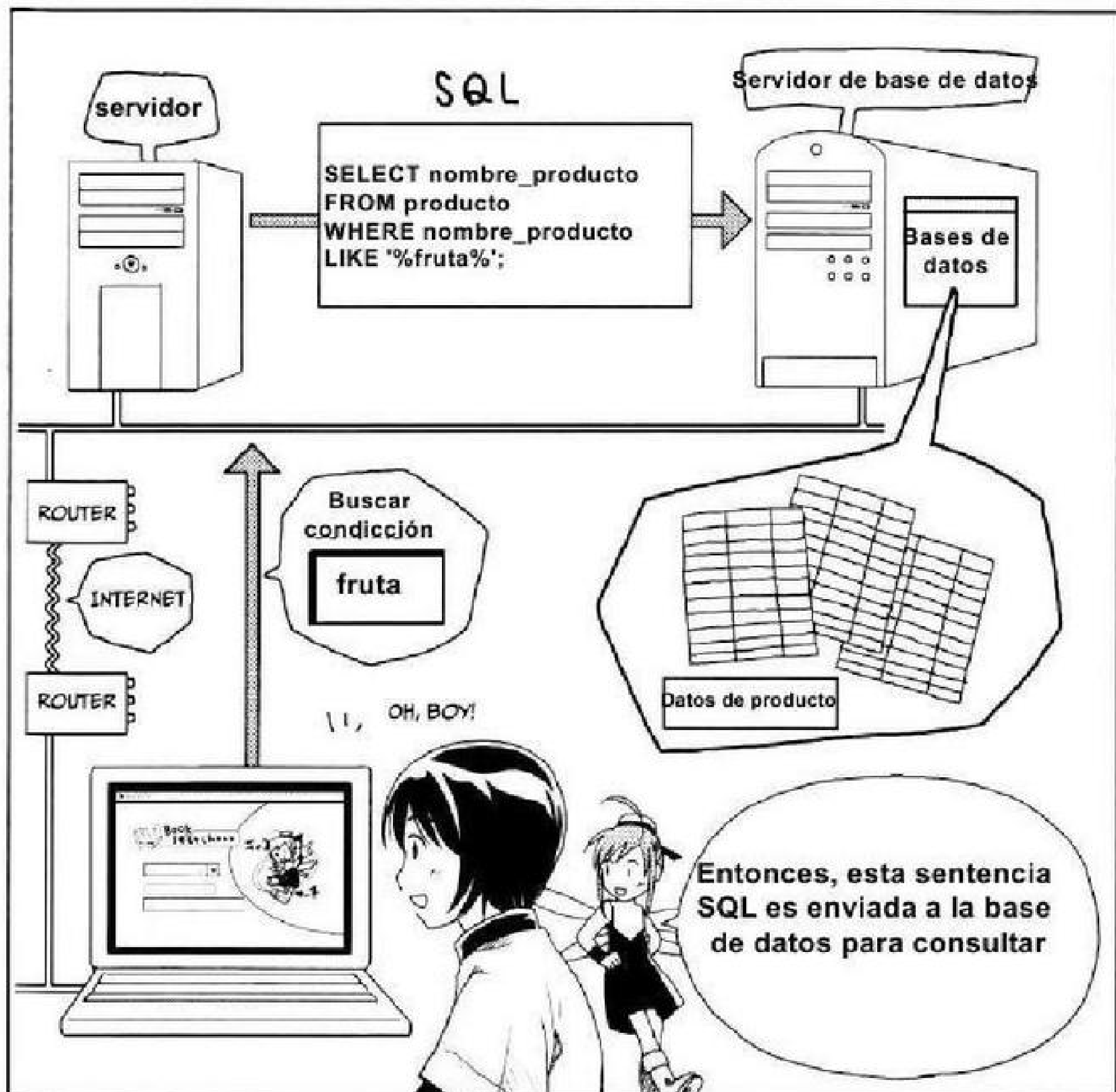
Book
search▶▶▶

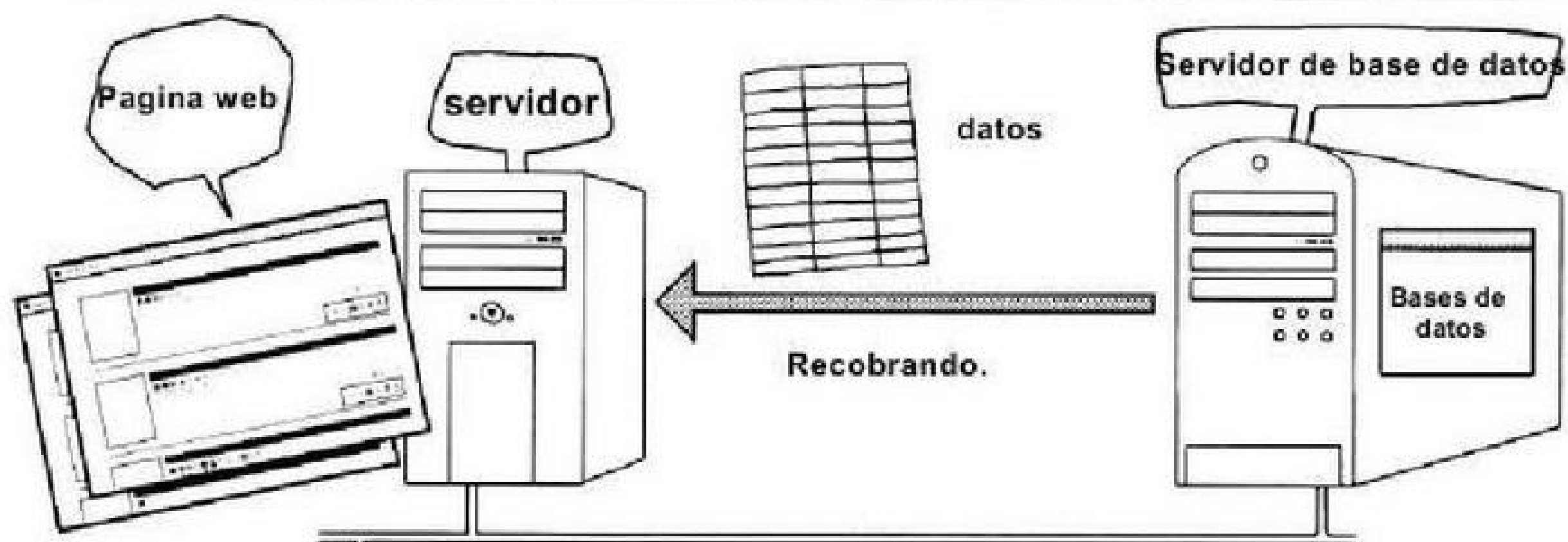
CATEGORIES ▼

KEYWORD

Fruit

Una sentencia de SQL
se realiza en el
servidor, supongo.







Me imagino que muchos
los clientes acceden a
una librería web al mismo
tiempo.

Incluso si el bloqueo
y la seguridad de las
funciones ofrecen la
protección completa
a la base de datos.



En ese caso, la
carga de procesos
son partidos entre
un número de
servidores.

¿Dices que más
que un servidor
es involucrado?



Si, la carga es distribuida entre
diferentes servidores, como un
servidor web y un servidor
de aplicación.



Un servidor web es
un servidor en cargado
de crear una pagina
web, ¿no?

¡Yep! Y un servidor de
aplicaciones es encargado
de la composición de la
sentencia SQL entre
otras cosas.



BASES DE DATOS DISTRIBUIDAS





PROCEDIMIENTOS ALMACENADOS Y DISPARADORES

En una red se debe en cualquier ambiente donde se utiliza un conjunto de servidores.

¡Correcto! Eso es cuando los procedimientos almacenados son útiles;

Ellos son a veces creados para ayudar a reducir la carga en la red.

¿Almacenado?

AHA!

No almacenado significa poner en la memoria, ¿en otras palabras?

¡Correcto!

En orden para reducir la carga en la red, frecuentemente son usadas operaciones que pueden ser almacenadas en una base de datos.

Frecuentemente son usadas las operaciones, dices.. que especie de operaciones son esas, ¿me asombro?

Bien, ya que estamos hablando sobre las operaciones para comprar un libro, sustrayendo de la tabla de inventario y agregando datos a la tabla de envío-

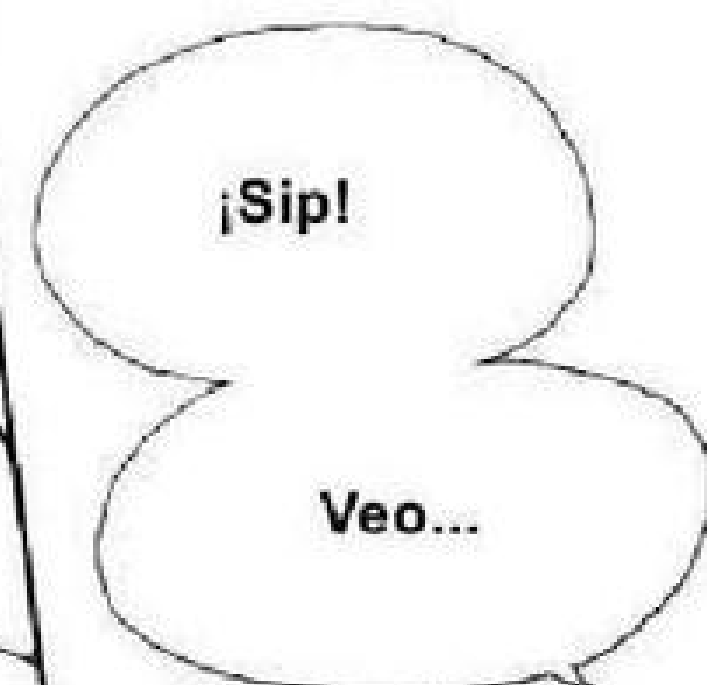
Vamos a ver...

¿No son esas operaciones típicas?











¡Una base de
datos de hadas!

びし

Simplemente voy
a decir hoy adios.

んんん
SHY
GIGGLE

Pero de alguna
manera he hecho
mas que eso, a
pesar de mi misma.

Ha sido corto
pero el tiempo fue feliz
con ustedes dos.



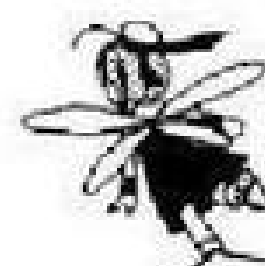
Tico...

Tico, ¡espera!

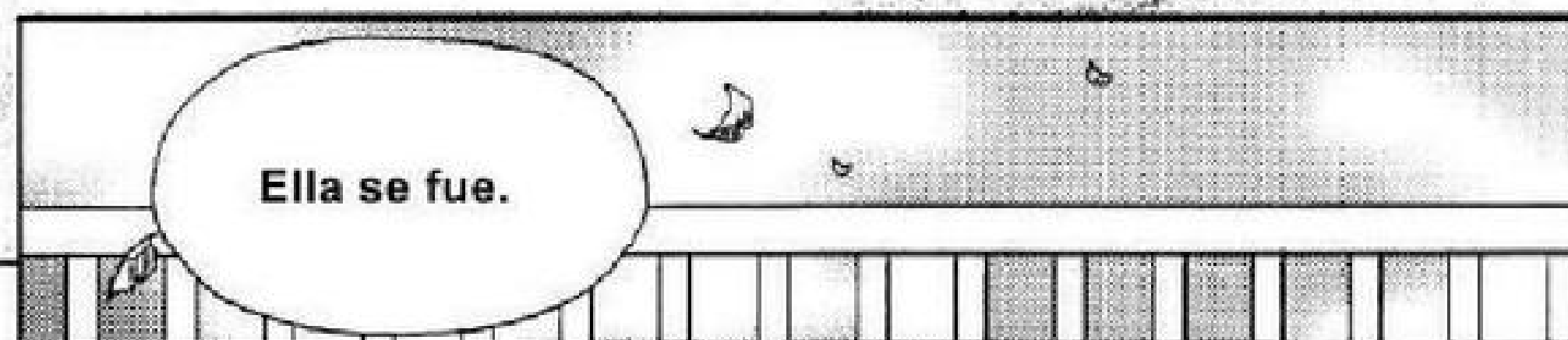
Yo tengo...



¡Querida, Tico!



¡¡Tico!!



Los días han pasado ..



¿Princesa todo esta bien con tu libro de base de datos?



¡Sí!

Estoy haciendo cosas faciles para que todos entiendan.

¿Quiere echar un vistazo?

¡Claro!



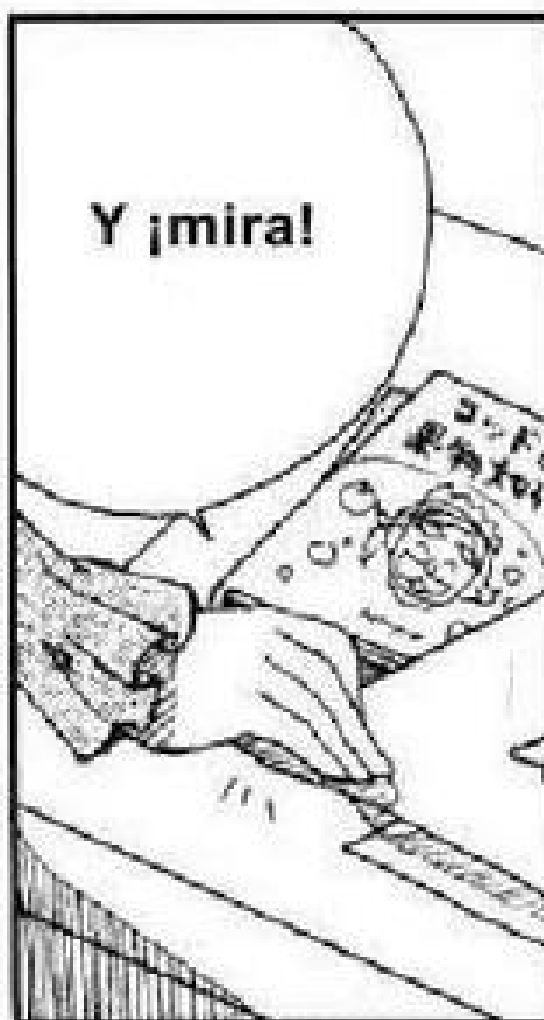
Es una buena idea hacer un libro con estilo comic.

Muy lindo.. ♡

Y Caín dibuja exelente.



Y ¡mira!



¡Aquí! Esto es la cubierta de lantera.

¡Fabuloso!





Hablando de Caín, el rey esta esperandote para hablar sobre el arreglo de la boda.

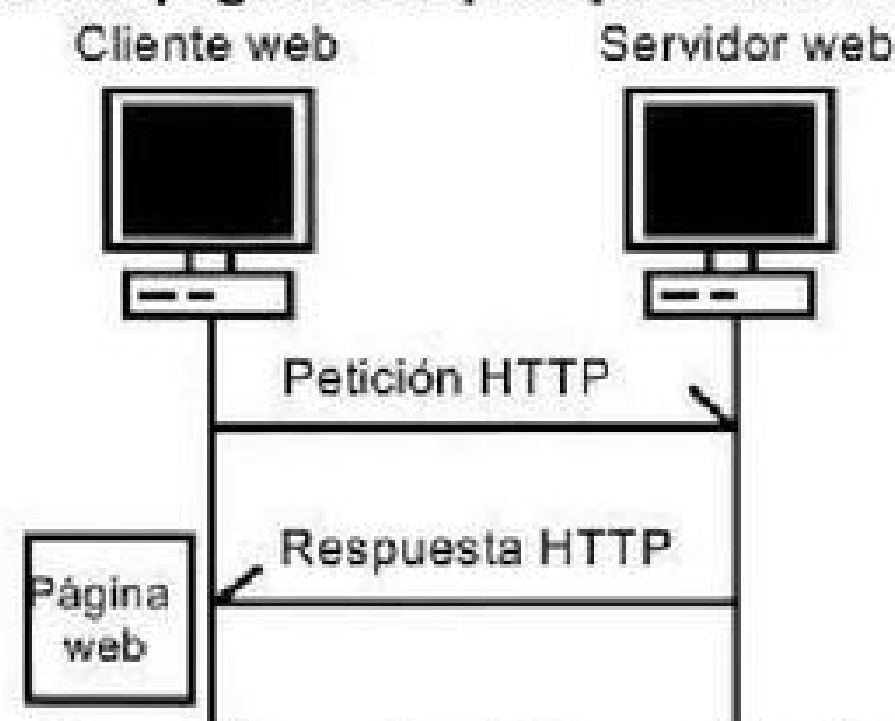


Fin

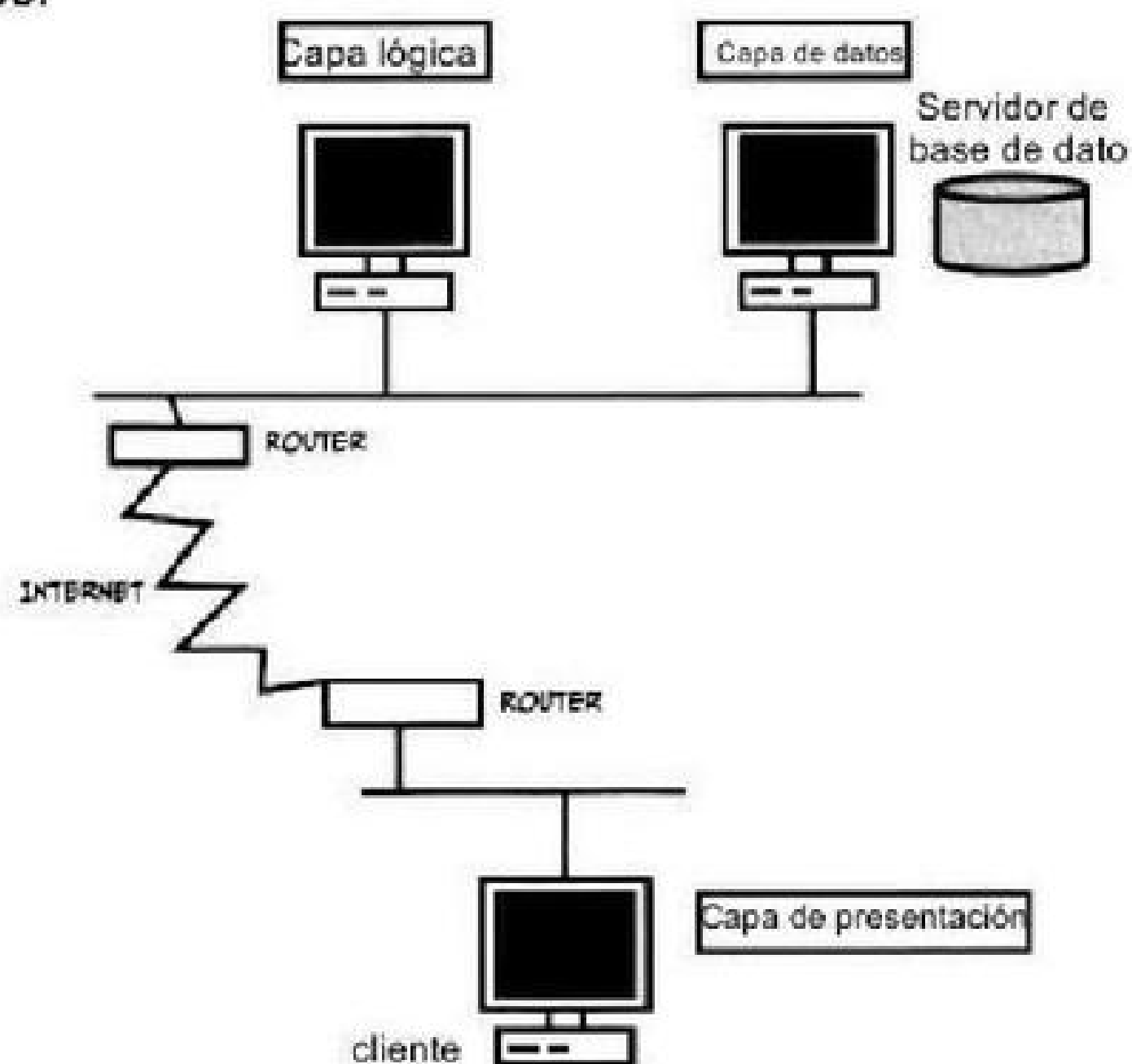
BASES DE DATOS EN LA WEB



Bases de datos son utilizadas para muchos propósitos diferentes, tales como sistemas de reserva de asientos de trenes y sistemas de depósitos bancarios. Son indispensables en la vida diaria y en las operaciones comerciales. Como mostré Ruruna y Caín, basado en la web los sistemas de base de datos son popular. En un sistema basado en la web, el protocolo de comunicaciones utilizado es Hyper Text Transfer Protocol (HTTP). software del servidor que se ejecuta en un servidor Web espera por una petición de un usuario. Cuando una solicitud de usuario (petición HTTP) es enviada, el software responde la petición y devuelve una página web correspondiente (respuesta HTTP). Una *página web* se compone de archivos de texto en formato HTML. Otros archivos especificados por Localizadores Uniformes de Recursos (URLs) se insertan dentro de una página web para presentar información como imágenes.



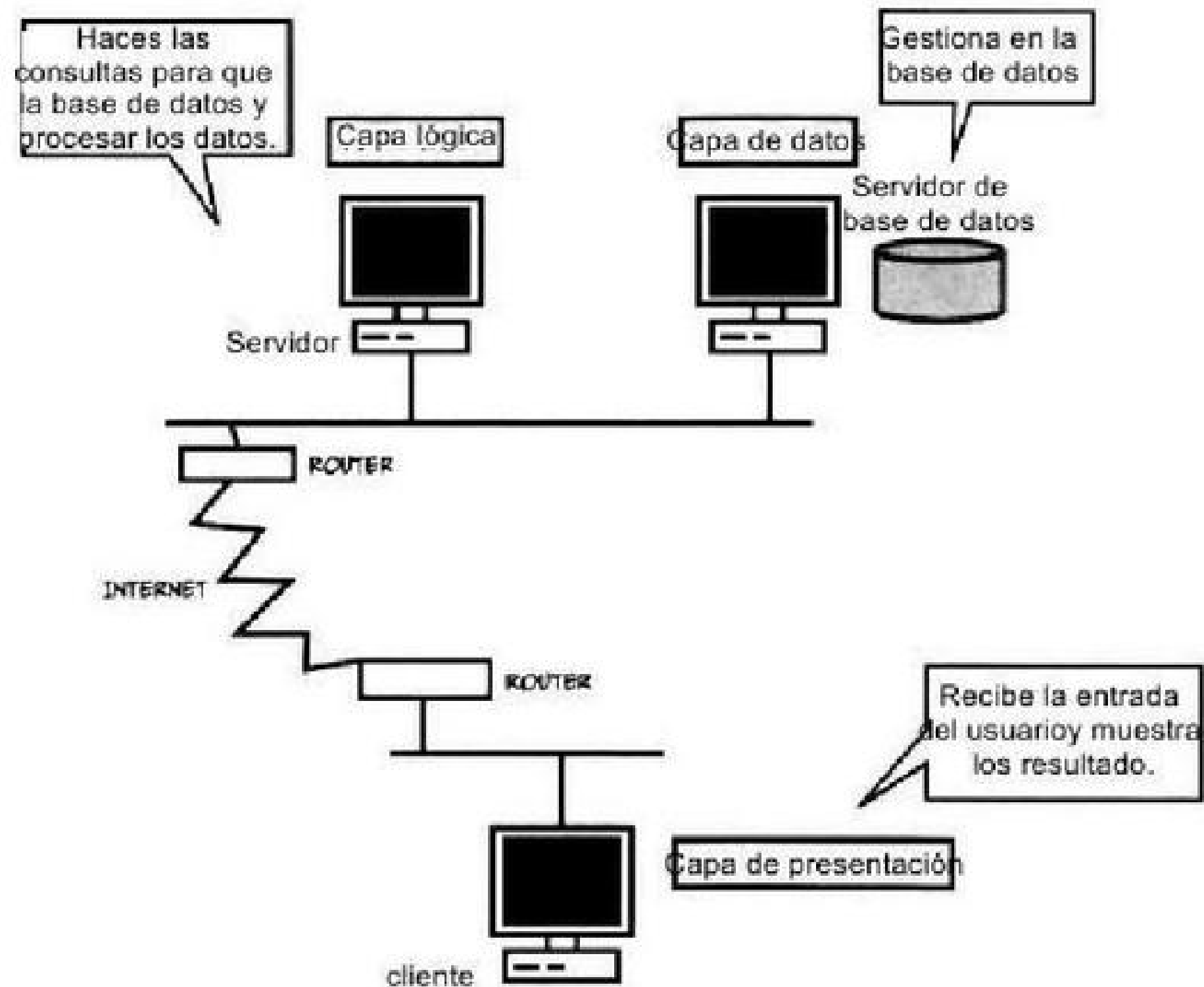
Cuando una base de datos es usada con una página web, un servidor de base de datos es añadido al sistema mostrado arriba. Este sistema puede ser configurado en tres capas y es relacionado como uno de tres niveles del sistema cliente / servidor. Un sistema de tres niveles cliente /servidor consiste de una capa de presentación, una capa de lógica y una capa de datos.



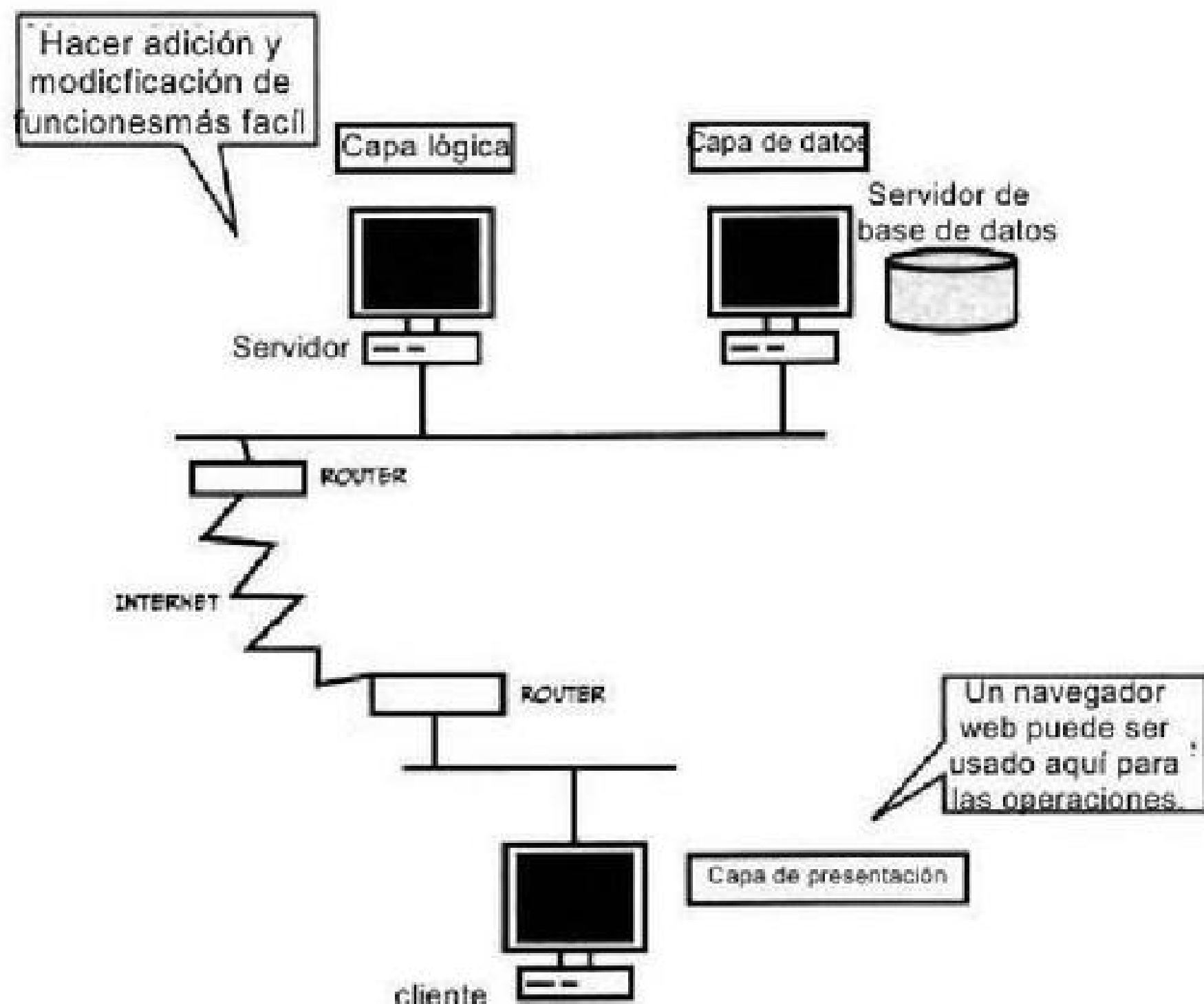
La capa de presentación recibe las entradas de los usuarios, tal como la búsqueda de condiciones, que necesitas para pasar a la base de datos. La capa de presentación también procesa el resultado de la consulta recibida de la base de datos de manera que puede ser mostrada. Un navegador web (tal como Internet Explorer o Firefox) funciona como herramienta de presentación para el usuario.

La capa lógica realiza los procesos de datos. Esta capa es donde las sentencias SQL son compuestas. Los procesos realizados aquí son escritos en uno o mas lenguajes de programación. Dependiendo de la satisfacción y carga de los procesos, varios servidores, tal como servidor de aplicaciones y un servidor web, puede ser usado para manejar los procedimientos.

La capa de datos procesa los datos en el servidor de la base de datos. Busca resultados que son devueltos de la base de datos en respuesta a las consultas SQL.



La configuración del sistema de tres niveles cliente/servidor es plexible y simple. Por ejemplo, cuando hace adiciones o modificaciones para una aplicación puedes separar la porción que quieres editar como una capa lógica. En la capa de presentación puede usar un navegador web, eliminando la necesidad instalando un programa software separador.



USANDO DE PROCEDIMIENTOS ALMACENADOS

En los sistemas basados en la web, también el mucho tráfico en la red puede ser un problema. Afortunadamente, puedes almacenar programas lógicos dentro del servidor la base de datos él mismo como almacén de procedimientos.

Almacenando procedimientos en el servidor de base de datos ayuda a reducir la carga en la red, por que elimina la necesidad de transferencia frecuente de consultas SQL. En añadir, almacenar procedimientos además hace más fácil el desarrollo de aplicaciones, ya que los procesos estándar pueden ser encapsulados en fácil de usar procedimientos. Actualmente, los procedimientos almacenados son una especie especial de una más amplia categoría llamada *programas almacenados*. Los otros dos tipos de programas almacenados son *funciones almacenadas* y *disparadores*.

Tipos de programas almacenados.

programa	Definición
Procedimientos almacenados	Programa que no devuelve valores de la elaboración de procesos.
Funciones almacenadas	Programa que devuelve valores de la elaboración de procesos.
Disparador	Programas que lanza automáticamente antes y después de las operaciones de bases de datos.



PREGUNTAS

¿Puedes responder estas preguntas? Las respuestas correctas están en la página 205.

Q1

En el sistema de tres niveles cliente/servidor. ¿En qué capa se opera la base de datos?

Q2

En el sistema de tres niveles cliente/servidor. ¿En qué capa recibe la interacción los usuarios y son mostrados los resultados?

¿QUE ES UNA BASES DE DATOS DISTRIBUIDA?



En un sistema basado en la web, la elaboración es distribuida entre un servidor de base de datos, y un navegador web, con diferentes grupos asignados para cada uno. Este tipo de sistema de distribución permite la elaboración flexible y disminuye la capacidad de elaboración requerida por cada servidor.

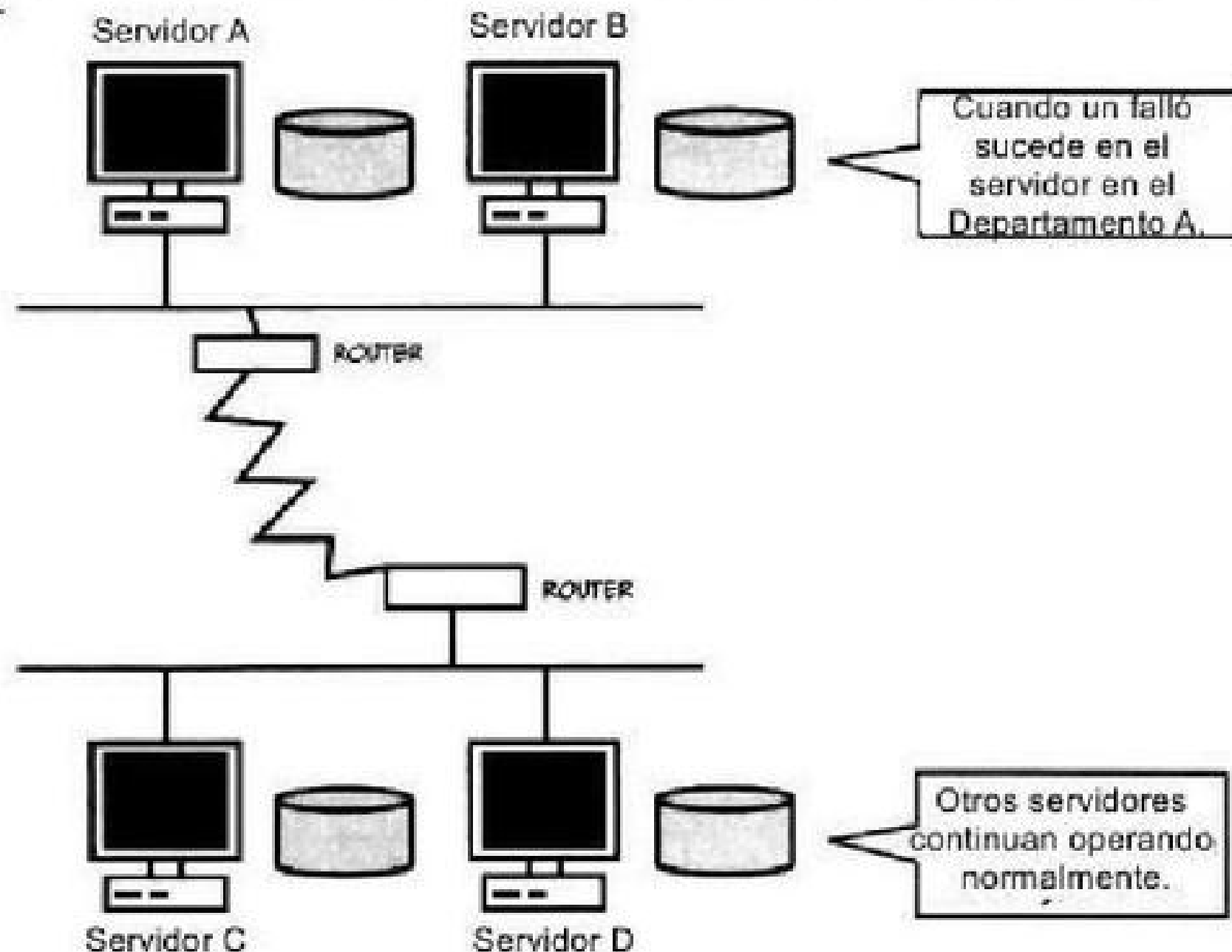
Pero un servidor de base de datos él mismo puede distribuir entre varios servidores. Distribuir los servidores de base de datos pueden estar en diferente localización o en la misma red. Nota. Jamás, que una distribución de base de datos puede ser manejada como una solo base de datos. Si la distribución de base de datos aparece como un solo servidor, el usuario no tiene preocupación sobre la localización de los datos o la Transferencia.

Una base de datos puede ser distribuida horizontalmente o verticalmente como lo verás.

DISTRIBUCIÓN HORIZONTAL

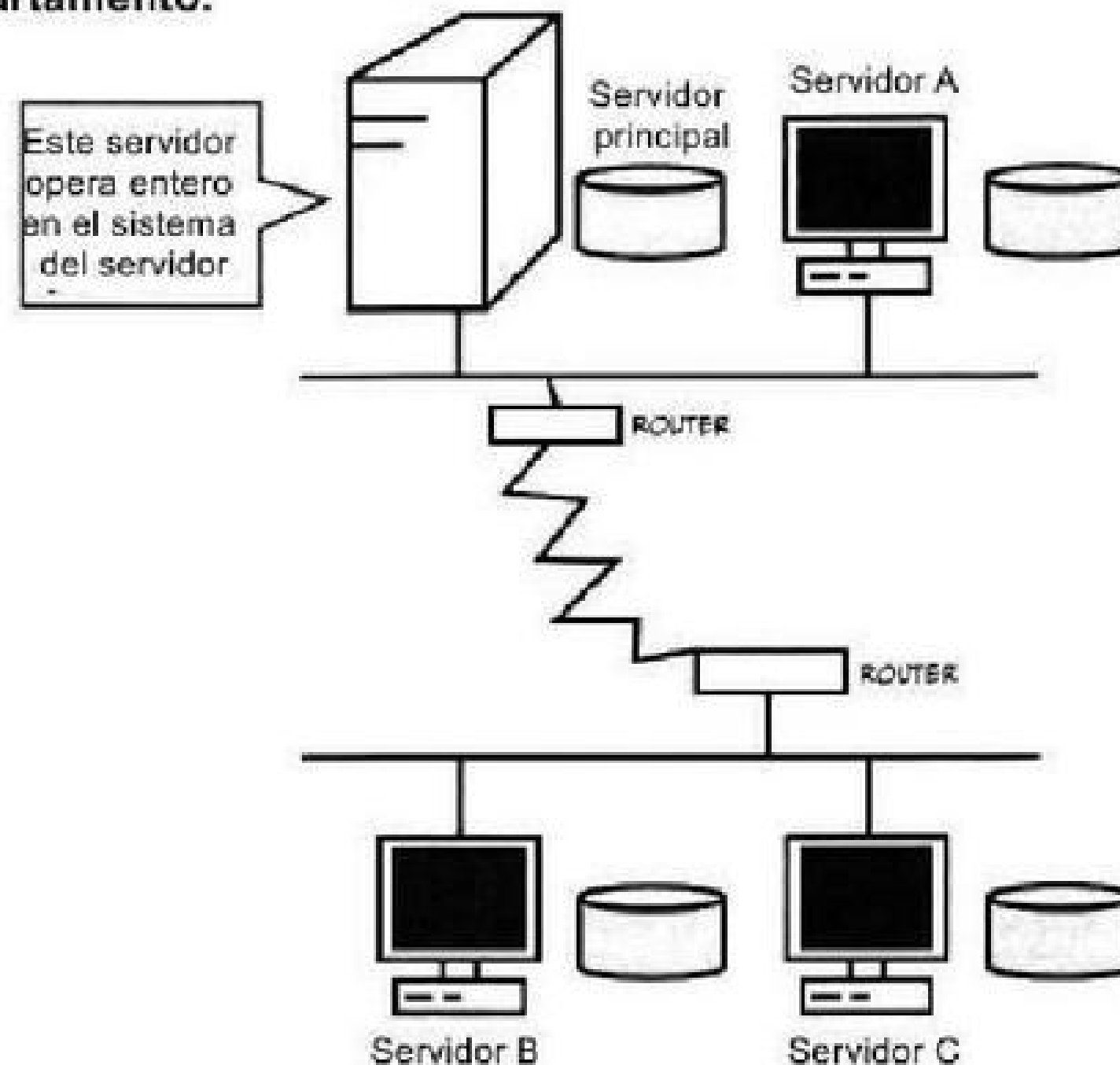
La distribución horizontal se usa en varios servidores de base de datos iguales. Cada servidor de base de datos puede usar datos de otro servidor de base de datos y en turno, cada uno hace lo mismo esta disponible para el otro servidor de base de datos. Esta estructura es usada para sistemas de extensas bases de datos que operan separadamente en cada departamento.

Una distribución horizontalmente de base de datos es un sistema resistente al fracaso por su diseño, entonces el fracaso en uno de los servidores no afectará las operaciones de la base de datos.



DISTRIBUCION VERTICAL

A la distribución vertical de servidores de bases de datos se le asignan diferentes funciones. Una de las funciones del servidor como servidor principal y realiza un rol de clave. En el momento que otros son en cargados de elaboración de tareas solicitadas. Una distribución verticalmente de base de datos hace más fácil el manejo del servidor de base de datos principal. Aunque este servidor principal tendrá una carga pesada. Un ejemplo de distribución vertil debería incluir una gran compañía del servidor principal y los servidores individuales que estan operando en cada departamento.



PARTICIONAMIENTO DE DATOS

En una distribucion de base de datos, los datos son extendidos a lado de los servidores para almacenar. Debes cuidadosamente considerar como son divididos los datos. Los datos pueden ser divididos de las siguientes maneras.

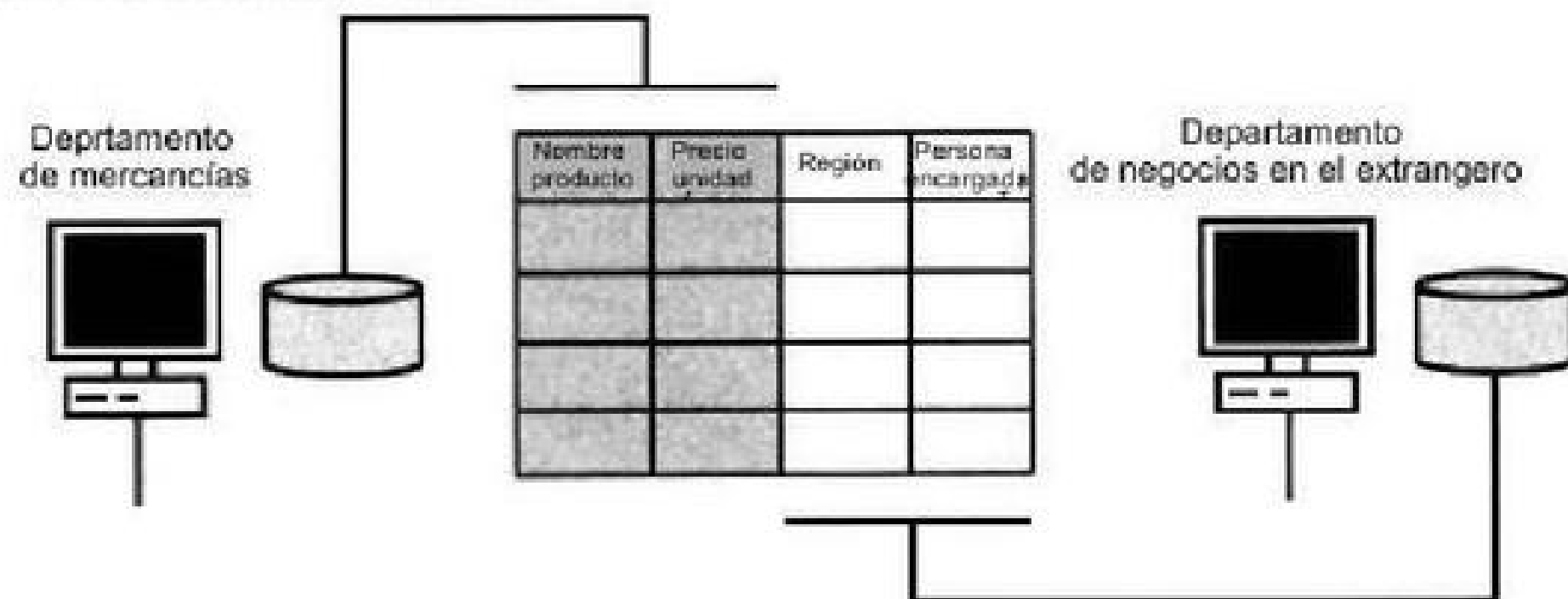
PARTICIONAMIENTO HORIZONTAL

Una *particion horizontal* divide los datos en unidades de filas. Filas resultando de la división son distribuidas a lado del servidor. Esta forma de particionamiento es con frecuencia usada cuando los datos pueden ser ordenados en grupos de tal manera que realacione los datos, qué son con frecuencia accedidos al mismo tiempo, son almacenados en el mismo servidor.



PARTICIONAMIENTO VERTICAL

Una partición vertical divide los datos en unidades de columnas. Columnas resultantes de la división son distribuidas entre los servidores. Por ejemplo, una partición vertical puede ser gestionada y unir bases de datos independientes pertenecientes a departamento de mercancías, departamento de negocios en el extranjero y el departamento de exportación.

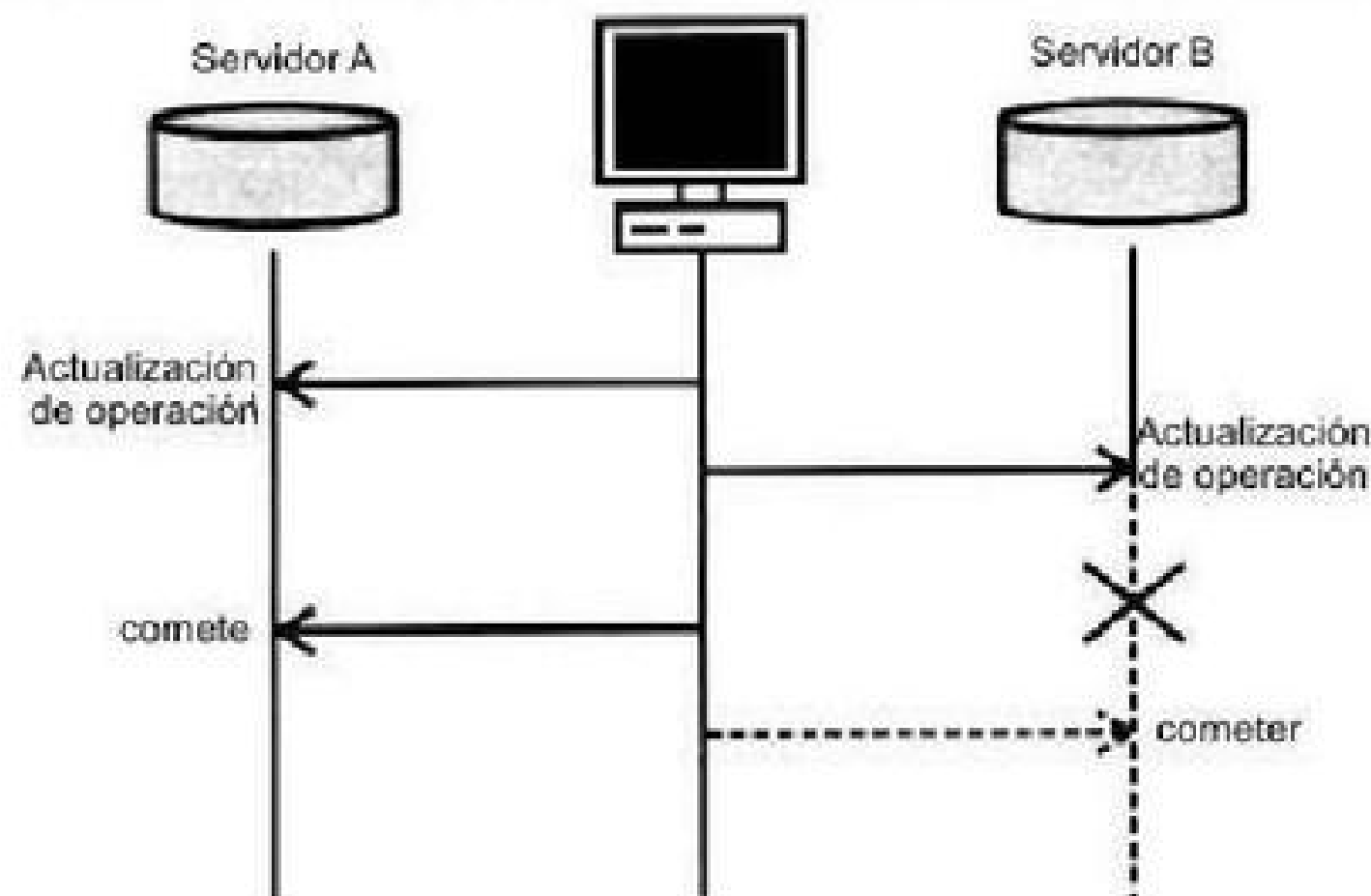


PREVINIENDO LA INCONSISTENCIAS QUE IMPIDEN COMETER EN DOS FASES



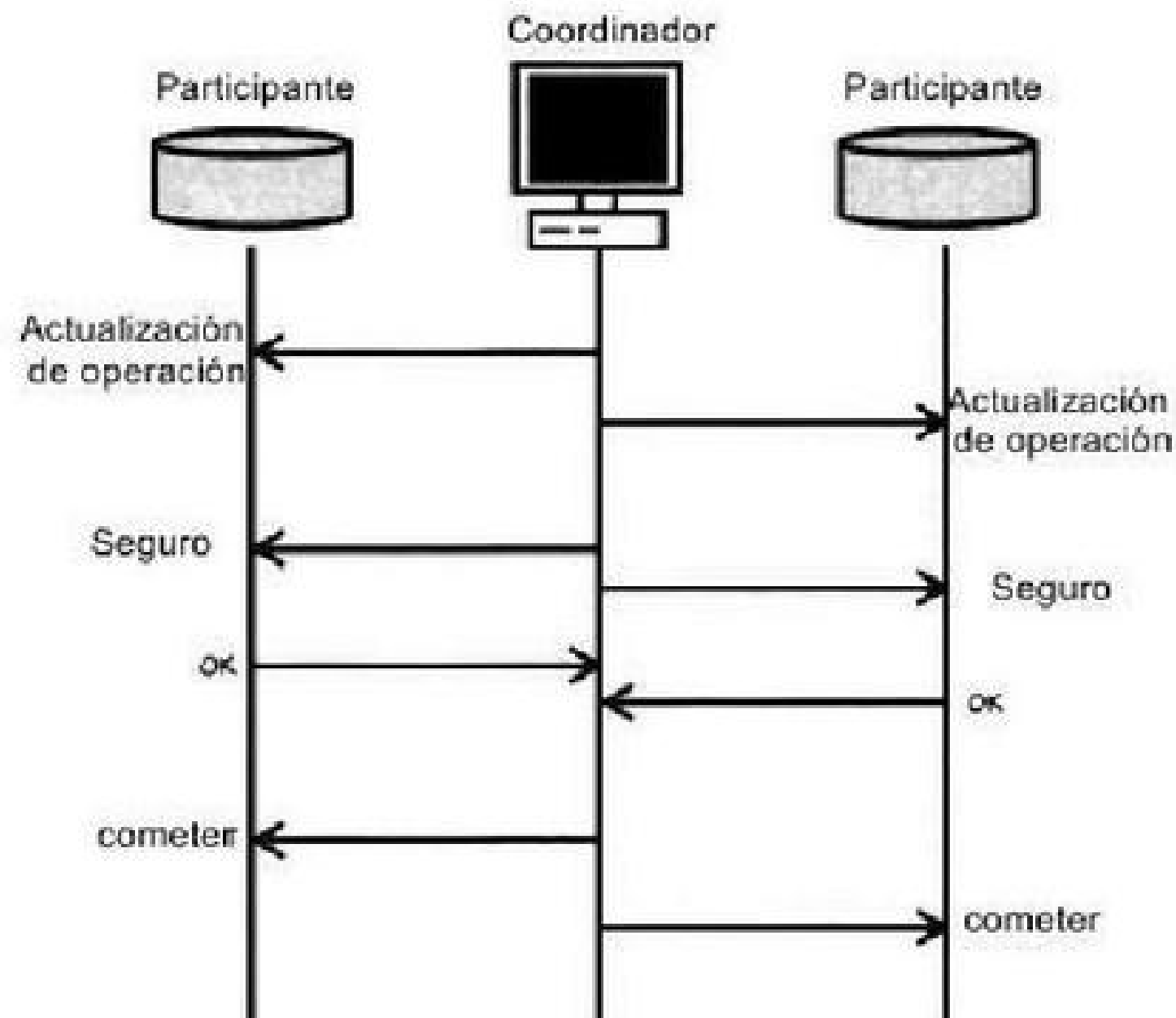
Una base de datos en diferentes servidores en un sistema de base de datos distribuida puede ser configurada para actuar como una sola base de datos en los ojos de los usuarios. Para lograr esto, varios pasos deben ser tomados para tratar con el hecho que los datos son actualmente distribuidos entre diferentes servidores.

Primero, siempre que los datos son comprometidos, todos los datos en todos los servidores deben ser actualizados consistentemente. En un sistema de base de datos distribuida, el método estándar puede cometer una ventaja para uno de los servidores inicia la actualización en otro momento no es como se muestra abajo. Esto es una violación de la propiedad atomicidad de transacciones, como esta transacción no finalizara con cualquiera de los dos que comete o reversión. Esto debe además causar en el sistema de base de datos un intento de favorecer la inconsistencia.

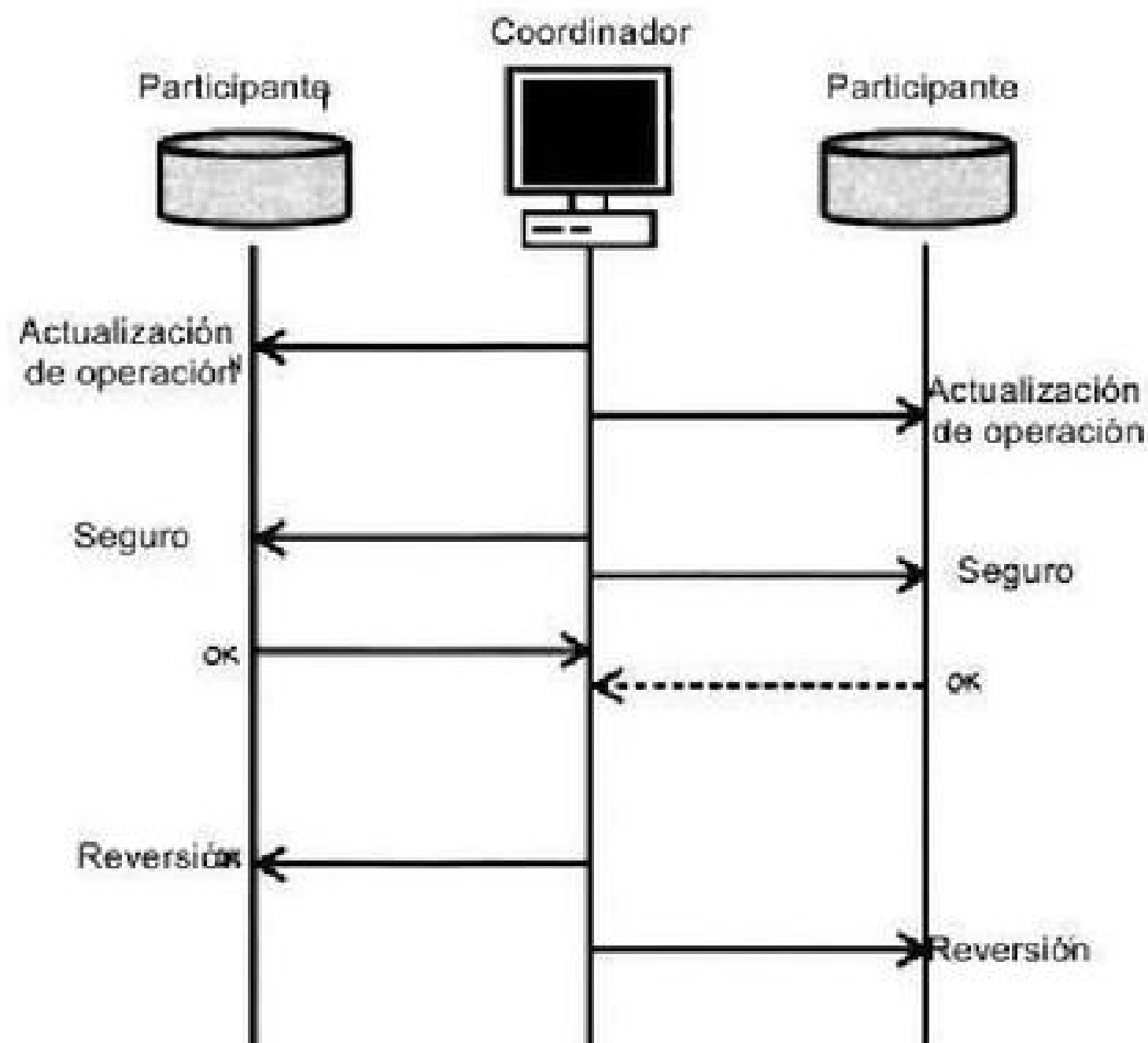


Por lo tanto, uno de dos fases cometida es adoptada en la distribución del sistema base de datos. Las dos fases cometidas crean una operación de cometido de ambos el primero y el segundo cometen operaciones.

Una de dos fases de operaciones cometidas implica un coordinador y participantes. En la primera fase de unde dos fases de operación cometida, el coordinador pregunta al participante si una operación cometida es posible. Los participantes envían en una respuesta OK si lo es. Este paso preliminar es referido como un preparado. En la segunda fase, el coordinador da las instrucciones para cometer, y todos los participantes realizan por consiguiente un cometido.



Si algun nodo falla la operación de seguro en cometer dos fases, todos los participantes reciben un directivo de reversión. Esto es como la base de datos en todos los servidores quedan consistentes con cada otro.





PREGUNTAS

Prueba estas preguntas sobre cometer dos fases. Las respuestas estan en la página 205

Q3

¿En un esquema de cometer dos fases, que instrucciones da el coordinador durante la rimera fase?

Q4

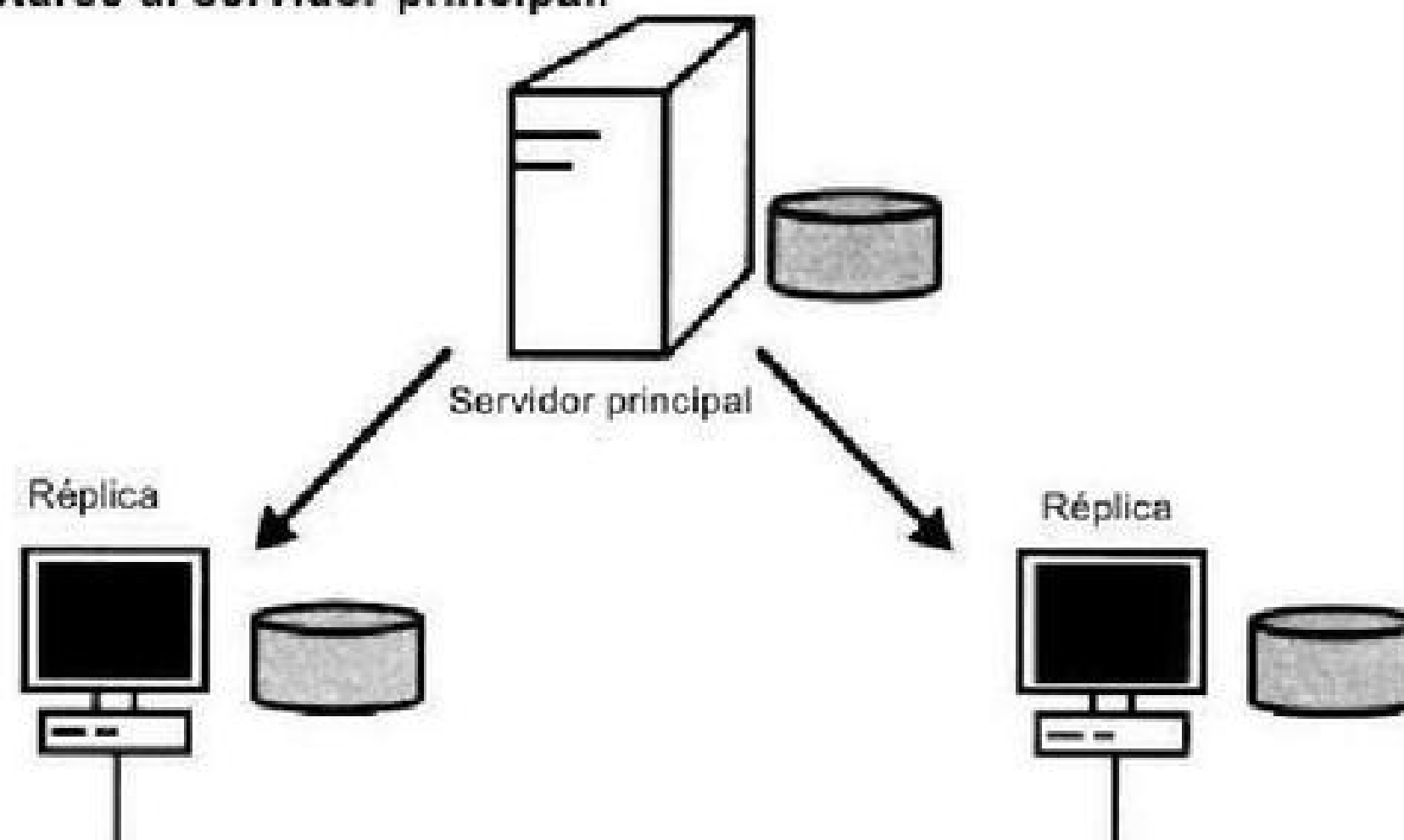
¿En un esquema de cometer dos fases que instrucciones da el coordinador durante la segunda fase?

REPLICACIÓN DE BASES DE DATOS

Algunas bases de datos distribuidas han duplicado o replicado, la base de datos reduce la carga en la red. Esta practica se refiere a como *replicación* La base de datos principal es referida como *base de datos maestra* y la copia es llamada la *réplica*. Hay varios tipos de replicación.

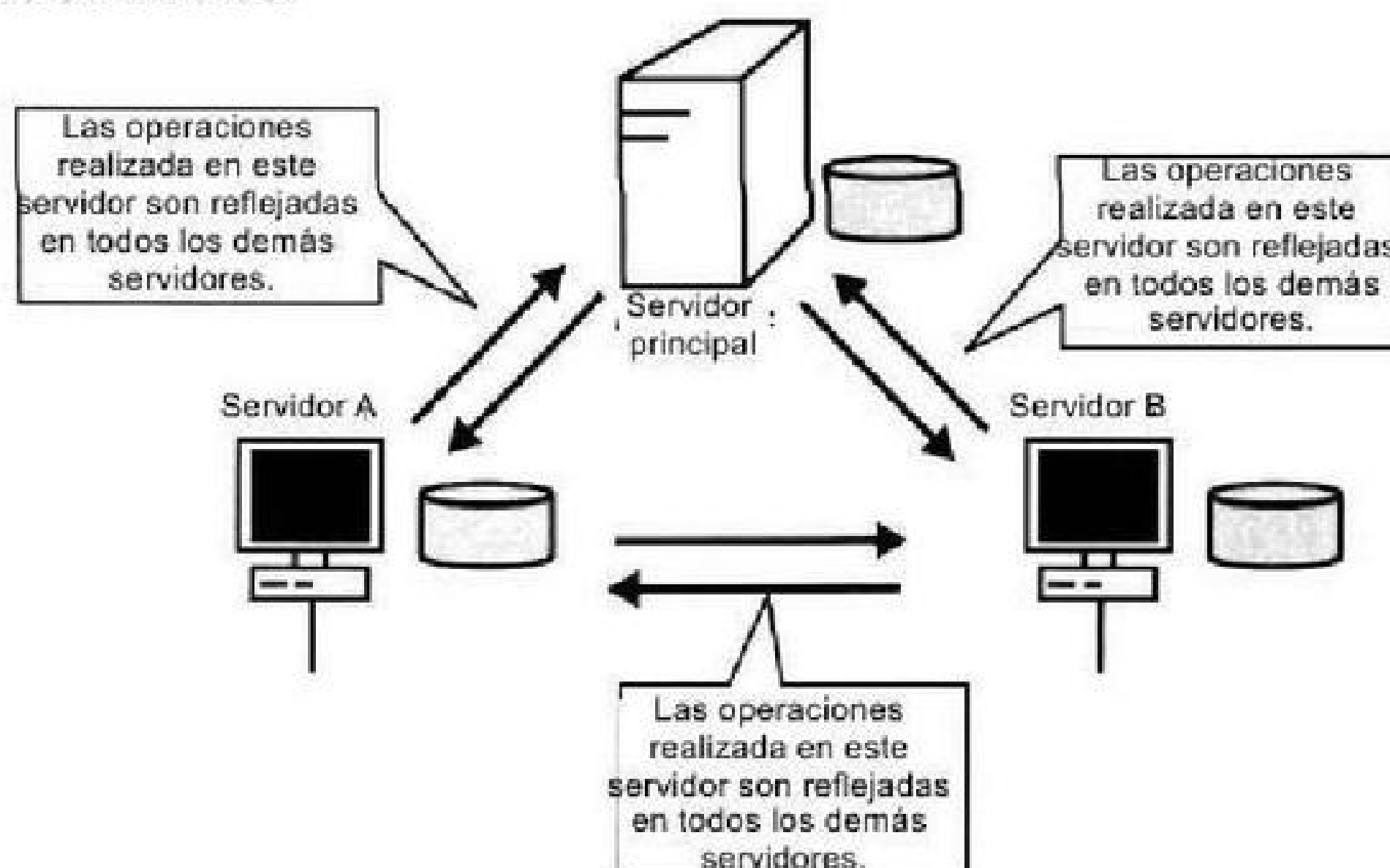
SÓLO LECTURA

Una réplica es sólo lectura creada y descargada de la base de datos maestra en el servidor principal. Para cambiar datos, los usuarios deben conetarse al servidor principal.



REPLICACIÓN HABILITADA PARA TODOS LOS SERVIDORES

En este método, el mismo maestro de base de datos es compartido por todos los servidores. Actualizaciones de algunos de los servidores son reflejadas en todos los otros servidores.



OTRAS APLICACIONES DE BASES DE DATOS

Esta es la sección final introduce la aplicación de tecnologías relacionadas a bases de datos.



XML

Lenguaje Marcado de Extensible (XML) está favoreciendo cada vez más la popularidad como un método de almacenamiento de datos. XML representa el acercamiento en las etiquetas. Desde que esas etiquetas pueden llevar información sobre los datos contenidos, este lenguaje es útil para almacenar datos y recuperarlos.

XML es útil porque su gramática es estrictamente estructurada hace los procesos programados fáciles. Por otra parte, XML viene en archivos de texto (que son fáciles de editar) y puede comunicarse con otros sistemas. Por esta razón, XML es usada a veces como un método de representación en lugar de base de datos.

```

<?xml version="1.0"?>
<productos >
  <fruta>
    <codigo producto>101</codigo producto>
    <nombre producto>Melon</nombre producto>
    <precio unidad>800</precio unidad>
  </fruta>
  <fruta>
    <codigo producto>103</codigo producto>
    <nombre producto>Manzana</nombre producto>
    <precio unidad>120</precio unidad>
  </fruta>
</productos>

```

BASES DE DATOS ORIENTADAS A OBJETOS

Una base de datos relacional almacena datos de texto en una tabla. Sin embargo, una base de datos relacional puede ser inadecuada cuando maneja ciertos tipos de datos. Que es donde una base de datos orientada a objetos (OOBD) viene.

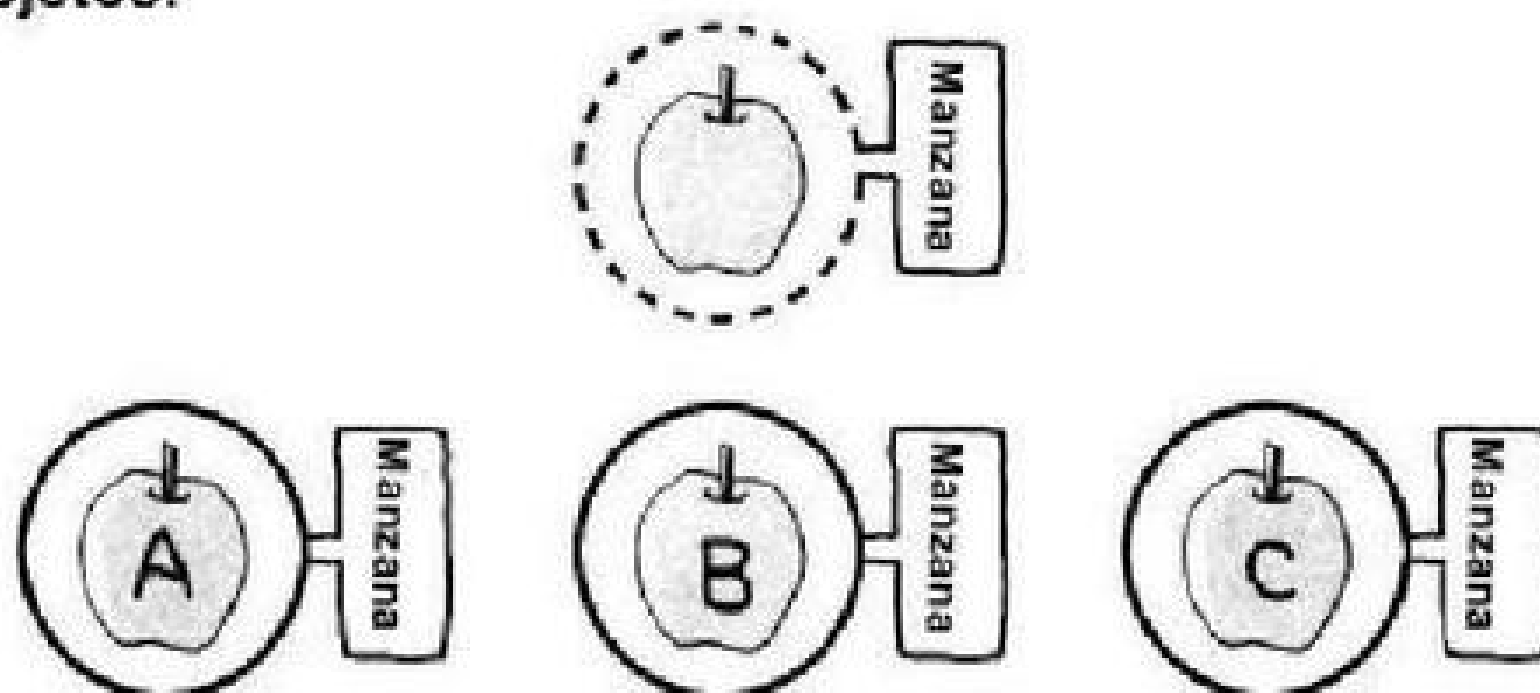
Los métodos orientados a objetos usan conjunto de objetos de datos e instrucciones en como que dato deben ser usados. Puede esconder los datos y solo exponer las operaciones sobre los datos en orden manejando los objetos como un componente independiente. Esta técnica se refiere a la *encapsulación*.

En una base de datos orientada a objetos, cada objeto es representado con un identificador. A veces, un objeto es solo llamado una *instancia*.

En una base de datos orientada a objetos, puedes solo manejar *un objeto compuesto* objeto anidado dentro de otro. Esto significa, por ejemplo, que puedas almacenar datos consistentes de una imagen combinada con texto como un solo objeto. Las bases de datos orientadas a objetos permiten la flexible administración de datos complejos.

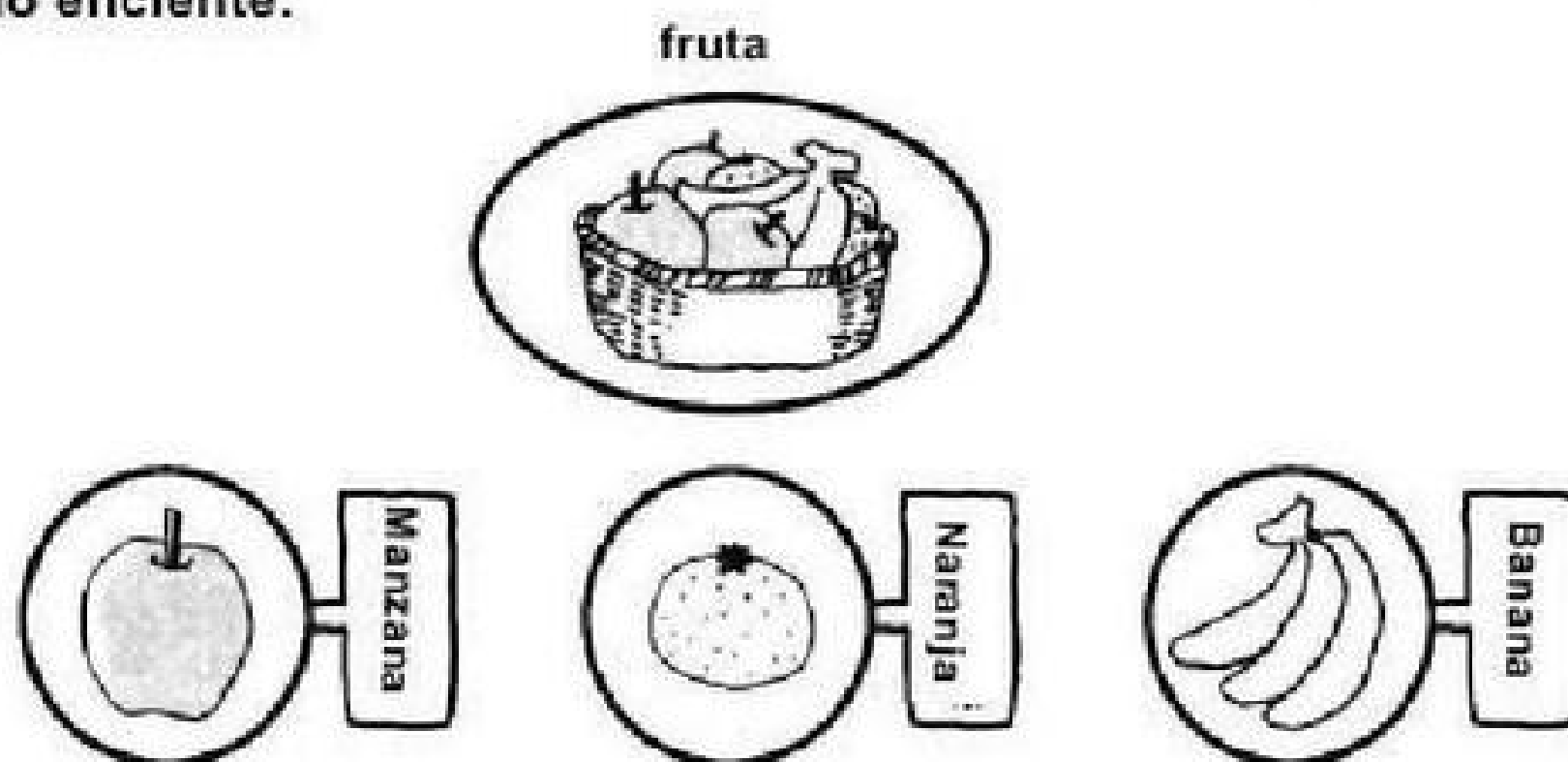


En una base de datos orientada a objetos, varios conceptos pueden facilitar el desarrollo orientado a objetos. La plantilla para los objetos es referida como una *clase*. Por ejemplo, supone que tiene que diseñar una clase Manzana. Objetos (instancias) en que la clase puede ser Manzana A, Manzana B, y así de este modo. La case Manzana habilita la creación de esos objetos.



En un esquema orientado a objetos, una clase solo puede relaciones jerárquicas. Puedes crear una clase hijo que tiene los mismos datos y funciones de una clase padre. Esta relación es referida como *heredada*. Puedes darle solo unicas funciones a la clase hijo.

Por ejemplo, la clase Manzana y la clase Naranja pueden heredar los datos y funciones de la clase fruta, pero ellas cada una tiene sus propios datos y funciones. En un esquema orientado a objetos puedes usar la relación jerárquica para permitir el desarrollo eficiente.



RESUMEN



- El sistema de tres niveles cliente/servidor es un método de sistema de configuración basado en la web.
- Una base de datos actúa como una capa de datos.
- Un sistema de distribución de base de datos maneja bases de datos que son dispersas.
- Un método de cometer dos fases es usado en una base de datos distribuida.

RESPUESTAS

Q1 Capa de datos.

Q2 Capa de presentación.

Q3 Preparar.

Q4 Cometer o reversión.

PALABRAS DE CLAUSURA

¿Has disfrutado estudiando bases de datos? Necesitarás aprender aún más después que puedas gestionar todos los aspectos de operación de una base de datos, pero lo fundamental de base de datos siempre permanecen igual. Firmemente comprendiendo lo básico, puedes identificar los datos significativos en el mundo real y diseñar y operar base de datos. Puedes adquirir experiencia avanzada de base de datos para construir tus propios conocimientos fundamentales. ¡Buena suerte!

SENTENCIAS SQL FRECUENTEMENTE USADAS.

CONSULTA BÁSCA

```
SELECT nombre_columna, ...  
FROM nombre_tabla;
```

CONSULTA CONDICIONAL

```
SELECT nombre_columna, ...  
FROM nombre_tabla  
WHERE condicion;
```

JUEGO DE PATRON

```
SELECT nombre_columna, ...  
FROM nombre_tabla  
WHERE nombre_columna LIKE 'patron';
```

BUSQUEDA CLASIFICADA

```
SELECT nombre_columna, ...  
FROM nombre_tabla  
WHERE condicion  
ORDER BY nombre_columna;
```

AGREGANDO Y AGRUPANDO

```
SELECT nombre_columna, ...  
FROM nombre_tabla  
WHERE condicion  
GROUP BY nombres_columnas_por_agrupar  
HAVING condicion_para_agrupar_filas;
```

UNIENDO TABLAS

```
SELECT nombre1_tabla.nombre_columna, ...  
FROM nombre1_tabla, nombre2_tabla, ...  
WHERE nombre1_tabla.nombre_columna = nombre2_tabla.nombre_columna
```

CREANDO UNA TABLA

```
CREATE TABLE nombre_tabla(  
nombre_columna1 tipo de dato,  
nombre_columna2 tipo de dato,  
....  
);
```

CREANDO UNA VISTA

```
CREATE VIEW nombre_vista  
AS SELECT sentencia
```

ELIMINANDO UNA TABLA REAL

```
DROP TABLE nombre_tabla;
```

ELIMINANDO UNA VISTA

```
DROP VIEW nombre_vista;
```

INSERTANDO UNA FILA

```
INSERT INTO nombre_table(nombre1_columna, ...)  
VALUES (valor1, ...);
```

ACTUALIZANDO UNA FILA

```
UPDATE nombre_tabla  
SET nombre_columna = valor1, ...  
WHERE condicion;
```

ELIMINANDO UNA FILA

```
DELETE FROM nombre_tabla  
WHERE condicion;
```

INDICE

A

Administración de datos no coordinados 10
Administración de datos separados 11, 19, 72
Administración de datos uniformes 19
Administración independiente de datos 19, 72
Agrupando 110, 159
Aislamiento 153, 155-158
Amortiguadores 161
Aplicación de servidores 182, 195
Atomicidad 153-154
AVG función (promedio) 98, 99, 110

B

B-árbol de indicación 163
Base de datos maestra 201-202
Base de datos
 Construyendo de un sistema existente 14
 Definido 6, 10, 15, 187
 Eficiencia de 3-4, 15, 19, 146, 174
 Tipos de 32-39
 Uso de 19-21, 175-182
Bases de datos en internet. Ver sistemas basados en la web
Bloqueando dos fases 156-157
Bloqueando granulación 157
Bloqueo exclusivo 134-136, 155-156
Bloqueo partido 133, 155-156
Búsqueda igual de llena 163

C

Campos 27-28, 30, 34, 35, 48
Campos únicos 30
Capa de datos 194-196, 205
Capa de presentación 205
Capa lógica 194-196
Carácter string 84, 108
Cardinalización 74
Clasificación. Ver funciones de agregación
Índices/indexación
 Columnas 34, 84
 Comentarios 30-21
 Comodines 97, 108
Condición actual 52, 55-56, 59, 74
Conjunto de funciones 98, 110
Conjunto de operaciones 39-42
Consistencia 153, 154, 155, 184
Consultas. Ver SQL sentencias SQL
Contar el acceso a disco 145

Contraseñas 141
Control 161-162
Controlando el acceso de los usuarios 19, 106, 126, 129, 141-142, 159-160, 167
Controles de bloqueo/basado en bloqueo 131-137, 155-157, 167, 175-176, 182
Controles coincidentes
 Control de estampillas 158
 Controles basados en bloqueo 131-137, 155-157, 167
 Controles optimistas 158
 Niveles de aislamiento 158
Controles de estampillas 158
Controles optimistas 158
Copias de respaldo 161
Correcto acceso 19, 106, 126-129, 141-142, 159-160, 167
Costo basado en los procedimientos 167
Creando vistas 117, 160

D

Datos corruptos 20, 154
Datos de seguridad 19, 127, 138-142, 159-160, 161-164, 167, 176, 182, 184
Datos duplicados 11, 16, 18, 19, 21, 29
Datos en conflicto 13, 17-18, 21, 60, 71, 116, 153, 158
Datos inconsistentes 153-154, 159, 199-201
Datos no autorizados sobrescritos 140
DBMS (sistema de administración de base de datos) 21
DCL (lenguaje de control de datos) 106
DDL (lenguaje de definición de datos) 106
Diseñando base de datos 19, 26
 Determinando condiciones de datos 74
 Modelo E-R (Entidad-Relación) 50-55, 74-77, 81
 Normalización 60-72, 78-81
 Pasos para 81, 84
Disparadores 187, 196
Distribución horizontal 197
Distribución vertical 198
Dividiendo tablas. Ver normalización
División de operaciones 37, 43, 45
DML (Lenguaje de Manipulación de Datos) 106
Durabilidad 153, 159-160

E

Elaboración de datos 35-37, 47-48, 130, 159, 167, 182, 195-198
Encapsulación 203
Entidades 52-54, 74

Entrada de datos 21, 90-92, 103-104, 106, 116
E-R modelo entidad-relación 50-55, 74-77, 81
Esquema conceptual 81
Esquema externo 81
Esquema interno 81
Esquemas 81
Etiquetas de datos, 202

F

Fallos de base de datos 161
Fallos de bases de datos 161
Fallos del sistema 161
Fallos medio 161
Filas 34, 84, 116
Formas 62-70, 81-82
Formas no normalizadas 62, 78-79
Frase GROUP BY 110
Frase HAVING 111
Frase WHERE 94-97, 106, 110, 119
Función COUNT 99-100, 110
Función hash 167
Función MAX (valor máximo) 99-100, 110
Función MIN (valor mínimo) 99, 110
Función SUM 99, 110
Funciones almacenadas 196
Funciones de agregación 98-100, 110-111

G

Granulación 157
Granulación alta 157
Granulación fina 157

H

HTML lenguaje de marcado de hiper texto 194
HTTP protocolo de transferencia de hiper Texto 178, 180, 194

I

Indexación hash 163
Índices/indexación 143-147, 162-164, 167
Instancias 203
Introduciendo datos 21, 90-92, 103-104, 106, 116
ISBN (estándar internacional de números de libros), 45
ISO (organización internacional para la estandarización) 124

J

JIS industria japonesa de estándares 124